Particle Methods in Bluff Body Aerodynamics

Rasmussen, Johannes Tophøj

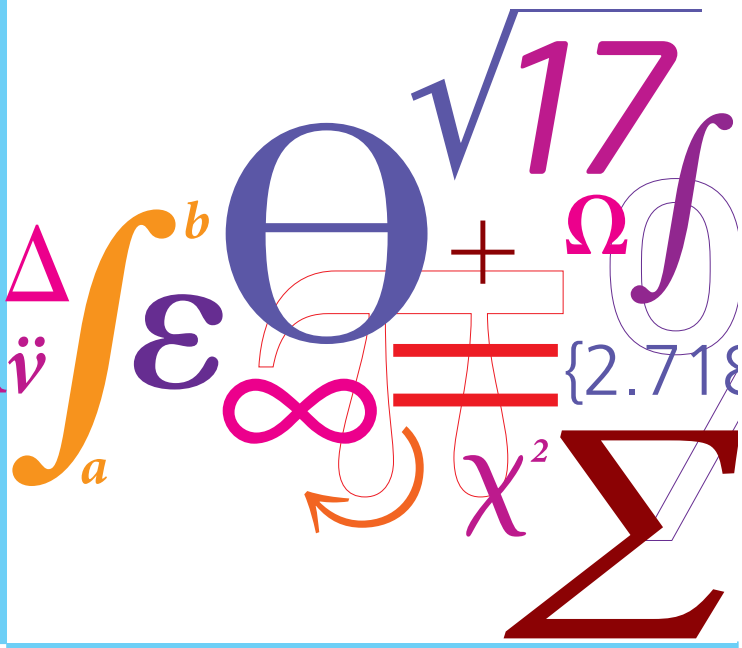
Publication date:
2011

Document Version
Publisher's PDF, also known as Version of record

Particle Methods in Bluff Body Aerodynamics

PhD Thesis

$$(EIv'')'' = q - \rho A \ddot{v}$$



Johannes Tophøj Rasmussen
DCAMM Special Repport No. S146
October 2011

DTU Mechanical Engineering
Department of Mechanical Engineering

Particle Methods in Bluff Body Aerodynamics

Johannes Tophøj Rasmussen

A dissertation submitted for the degree of Doctor of Philosophy

Section of Fluid Mechanics
Department of Mechanical Engineering
Technical University of Denmark

October 2011

Preface

This dissertation is submitted in partial fulfilment of the requirements for obtaining the Ph.D. degree in mechanical engineering at the Technical University of Denmark (DTU). The Ph.D. Project was funded by the Danish Research Council of Independent Research (Grant. No. 274-08-0258) and carried out at the Department of Mechanical Engineering (MEK), section of Fluid Mechanics at DTU in the period September 1st 2008 - November 1st 2011. Supervisor on the project was Associate Professor Ph.D. Jens Honoré Walther from the department.

There is much and many people to appreciate. In the context of the last three years of Ph.D work at DTU: First and foremost I owe a special thanks to my tough but supportive 'boss' Jens Honoré Walther from whom I have learned very much. For making me 'keep digging' and keeping me on track. I am very grateful for being able to work with you. For our collaboration I would like to thank Mads M. Hejlesen, Allan Larsen and George-Henri Cottet. I would like to express my gratitude towards my co-workers in building 403, in particular the other Ph.D. students, for the pleasant atmosphere and the academic discussions. And Laust for our talks on vortices and beyond.

I would furthermore like to thank the following: Ole Christensen for his pointers on series expansions. Petros Koumoutsakos for my visit to the CSE lab at ETH. Ivo Sbalzarini and the MOSAIC group at ETH for our collaboration and in particular I would like to thank Omar Awile for his support on PPM. I would like to thank George-Henri Cottet and his team for the pleasant stay at Laboratoire Jean Kuntzmann at UJF. And of course Bennos Hvile for all the love.

In this dissertation no nomenclature is given but variables are declared when mentioned.

Parts of the work in the present dissertation have been published in *Journal of Wind Engineering & Industrial Aerodynamics* [84] and *Journal of Computational Physics* [83]. A third paper is being prepared [43].

Resumé (in Danish)

Interaktionen mellem strømninger og strukturer studeres ved hjælp af numeriske metoder både i industrielt og akademisk regi. Strømningers komplekse dynamik ved høje Reynolds tal kan ikke alene omsættes med rå beregningskraft, hvorfor det er nødvendigt med mere effektive og specialiserede beregningsalgoritmer. Denne afhandling fokuserer på brugen af hvirvelmetoder og effektiv udnyttelse af beregningsressourcer og er inddelt i tre emner.

Først præsenteres en ny metode til bestemmelse af den aerodynamiske admitans gennem numerisk simulering. Metoden bygger på generingen af turbulente hvirvelpartikler der indsættes opstrøms i strømningen. Turbulensen generes før afviklingen af simuleringen og bygger på atmosfæriske turbulensspektra og en rummelig korrelationsfunktion. Metoden er valideret ved simulering af turbulente strømninger forbi en flad plade og Storebæltsbroen, Øresundsbroen og Busan-Geoje broen.

Afhandlingen introducerer endvidere en ny vortex-in-cell algoritme med varierende rummelig opløsning. Poisson ligningen som beskriver forbindelsen mellem strømningens vorticitet- og hastighedsfelter løses ved hjælp af fast Fourier transforms i ubegrænsede domæner. Faste legemer påtrykkes ved hjælp af Brinkman penalisering i en semi-implicit formulering. Det vises at penaliseringen kan påtrykkes som en enkel interpolation. Implementeringen er todimensionel og sekventiel. Implementeringen valideres mod en analytisk løsning til Perlman vorticitetsfeltet, impulsivt startede strømninger omkring statiske og roterende cylindre samt strømning omkring en brosektion.

Til slut præsenteres en tredimensionel vortex-in-cell implementering som bygger på et eksisterende open source software-bibliotek som muliggører parallelle tredimensionelle strømningssimuleringer baseret på hvirvelpartikler. En højere-ordens Fourier-baseret Poisson-løser præsenteres og valideres. Ved hjælp af regulariseringsteknikker opnås formelt en arbitrært høj rummelig konvergensrate. Implementeringen er forberedt den udviklede algoritme til varierende opløsning hvilket dog ikke er færdigimplementeret i det understøttende softwarebibliotek. Simuleringen af deformerende og bevægende objekter er demonstreret ved hjælp af Brinkman-penalisering og vortex-in-cell algoritmen. Modellen er afprøvet på strømning omkring kugler, en brosektion i opførelsesfasen samt vandmanden *Aurelia aurita*'s fremdrift.

Abstract

Fluid-structure interaction is studied numerically in academics and the industry. Shear computational power alone is insufficient to accurately resolve the complex dynamics of high Reynolds number fluid flow. Therefore the development of more efficient and applicable computational algorithms is important. This dissertation focuses on the use of vortex particle methods and computational efficiency. The work is divided into three parts.

A novel method for the simulation of the aerodynamic admittance in bluff body aerodynamics is presented. The method involves a model for describing oncoming turbulence in two-dimensional discrete vortex method simulations by seeding the upstream flow with vortex particles. The turbulence is generated prior to the simulations and is based on analytic spectral densities of the atmospheric turbulence and a coherence function defining the spatial correlation of the flow. The method is validated by simulating the turbulent flow past a flat plate and past the Great Belt East bridge, the Øresund bridge and the Busan-Geoje bridge.

The dissertation introduces a novel multiresolution vortex-in-cell algorithm using patches of varying resolution. The Poisson equation relating the fluid vorticity and velocity is solved using fast Fourier transforms subject to free-space boundary conditions. Solid boundaries are implemented using the semi-implicit formulation of Brinkman penalization and it is shown that the penalization can be carried out as a simple interpolation. The implementation is two-dimensional and sequential. The implementation is validated against the analytic solution to the Perlman test case and by free-space simulations of the onset flow around fixed and rotating circular cylinders and bluff body flows around bridge sections.

Finally a three-dimensional vortex-in-cell algorithm is implemented into an existing open source library that enables large scale, three-dimensional particle-vortex simulations. A high order Fourier based Poisson solver is presented using novel use of regularization in the vortex-in-cell algorithm which formally enables arbitrarily high order convergence. The implementation is prepared for multiresolution though it is currently not supported by the parallel framework. The simulation of deformable and moving objects is demonstrated using Brinkman penalization and the vortex-in-cell algorithm. The model is applied to flow around spheres, a bridge section during the construction phase and the swimming motion of the medusa *Aurelia aurita*.

Contents

Contents

Introduction

Computational fluid dynamics (CFD) is a widely used tool for engineers designing structures exposed to fluid flow. By simulating the motion of fluids on computers we gain knowledge of their often invisible dynamics. In fact we may quietly study the entire flow field, advance it temporally as desired and compute the necessary statistical properties. Physicists and engineers alike use CFD to gain insight into the details of fluid phenomena beyond what is possible experimentally and analytically. This is the value of CFD; to create accurate observations and measurements from which improved design may be proposed or physical models may be identified. With regard to accuracy and time consumption CFD is competitive with experiments and provides valuable tools. The use and accuracy of computer simulations has limitations due to the complex physical processes and due to the large amount of information that makes up fluid flow. Therefore it might be necessary to include modelling of e.g. turbulence, multiphase flow, phase change, complex meshes, including mesh motion and morphology, or other models. The range of CFD algorithms and models has grown and improved, extending the use and applicability of CFD. Current vortex methods frequently used in bridge aerodynamic are limited to two-dimensional analysis. Far-field time dependent boundary conditions are difficult to model in the existing methods.

This dissertation addresses limitations of numerical flow modelling with an application to bridge aerodynamics in mind, where fluctuating forces are of great concern. These could be the forces induced by atmospheric turbulence and forces due to the interaction between the flow and the bridge. To fully capture the effects of vortex shedding the latter requires three-dimensional simulation and thus suitable parallel algorithms. The vortex-in-cell combined with Brinkman penalization techniques offer an efficient and straightforward tool to this end. The present work focuses on the performance of the vortex-in-cell algorithm and Brinkman penalization, aiming at massively parallel simulations. Furthermore a novel technique is demonstrated using vortex particle methods to numerically determine the aerody-

namic admittances; a statistical property otherwise found experimentally at limited range of turbulent scales.

Dissertation structure

Chapter 1 outlines the physics of interest; fluid-structure interaction and the parametrised forces and spectral transfer functions of interest in bridge aerodynamics. The chapter further introduces the family of numerical models that is based on vortex particles.

Chapter 2 describes the generation of turbulent velocity fluctuations in vortex particle simulations and the procedure to determine the aerodynamic admittance from simulations. The procedure is investigated and validated against analytic solutions and experimental results.

A multiresolution vortex-in-cell algorithm is introduced in chapter 3. The two-dimensional implementation is validated against analytic solutions to vorticity fields and drag and lift around static and rotating cylinders.

During the present work the vortex-in-cell has been implemented with a structure prepared for the presented multiresolution algorithm. A Poisson solver based on the fast Fourier transform has been implemented in the underlying parallel software library. Chapter 4 validates both the Poisson solver and the vortex-in-cell implementation.

Chapter 1

Fluid-structure interaction

Aerodynamic forces on structures can lead to destructive and catastrophic events. Strong winds may cause great destruction through powerful forces but more subtle phenomena may lead to structural failure even at relatively low wind speeds. Aeroelastic flutter is such a phenomenon where aerodynamic forces couple with the natural frequency of the structure and may lead to motion in a positive feedback loop. One of the best documented cases of aerodynamic flutter is the collapse of the Tacoma Narrows bridge in July 1940, only four month after opening, see Fig. 1.1. Simulations may be performed for optimisation during the design phase, to establish critical parameters or be used in academic work.

Flow simulations can be used to prevent error-prone structures by computing the time history of the aerodynamic forces while providing detailed visualisation of the flow field. E.g. the derivatives of aerodynamic coefficients, buffeting and associated flutter limit can be extracted from the simulations by imposing a prescribed heave and pitch motion of the bridge section [64, 108].

The present work focuses on numerical methods relevant for vortex shedding and aerodynamic forces, phenomena relevant for bridge engineering.

1.1 Vortex shedding

As flow passes solid objects boundary layer vorticity forms and builds up. Assemblies of vorticity is in turn advected with the flow from alternating sides of the solid. This is referred to as vortex shedding. Vortex shedding is dependent on the non-dimensional Reynolds number

$$\mathrm{Re} = \frac{UL}{\nu} \tag{1.1.1}$$

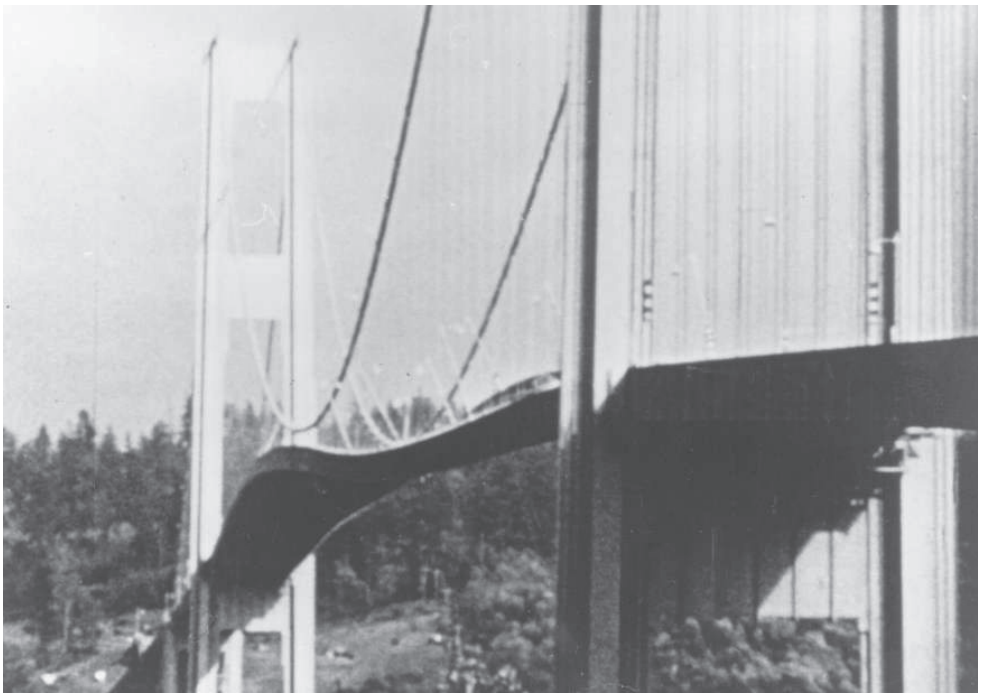


Figure 1.1: The first Tacoma Narrows bridge was strongly influenced by aeroelastic flutter. It collapsed the 1st of July 1940 just four months after opening.

Figure 1.2: The von Kármán vortex street behind a circular cylinder at Re=500. Visualisation of `vic20` simulation.

which relates the dynamic forces of the flow to the viscous forces. U is a characteristic flow velocity, in the present context often the far-field velocity, and L is a characteristic length. ν is the kinematic velocity. At low Reynolds numbers the flow around structures may stay attached to the bluff bodies but as the Reynolds number increases the flow will separate and oscillate [77]. The resulting oscillating forces are of interest for comparison with natural frequencies of the structure to limit vortex induced vibrations. The oscillating wake behind a cylinder is the well known von Kármán vortex street shown on Fig. 1.2. The Strouhal number

$$\mathrm{St} = \frac{fL}{U} \tag{1.1.2}$$

is the non-dimensionalised shedding frequency f. For circular cylinders $\mathrm{St} \approx 0.2$ in a wide range of Reynolds numbers [109].

The complexity of fluid flow increases with the Reynolds number but even at moderate Reynolds numbers flow around two-dimensional objects become three-dimensional [91]. However, two-dimensional simulations still provide good predictions of the parametrised force coefficients and are commonly used as an engineering tool [64, 59, 60]. The vortex shedding of two-dimensional simulations become disharmonic (see Fig. 1.3) contrary to experiments and three-dimensional simulation [98] and fail to accurately predict the shedding frequency [91, 98].

1.2 Aerodynamic Admittance

The influence of turbulence on the aerodynamic forces can be quantified by the aerodynamic admittance which is the of focus of parts of the present work. The aerodynamic admittance is a spectral transfer function relating

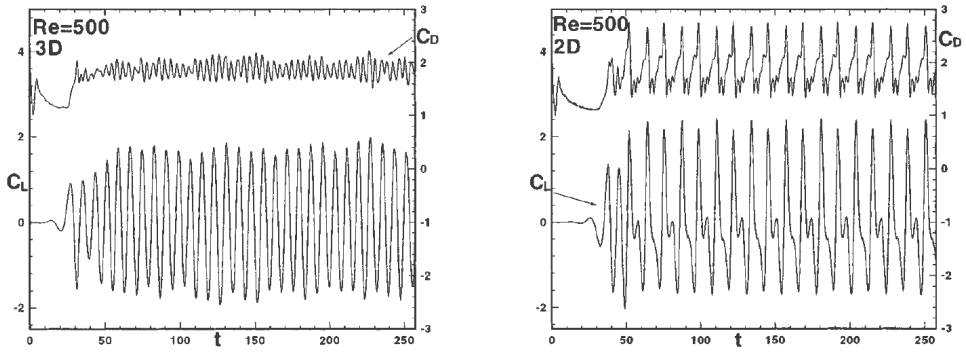


Figure 1.3: Lift and drag around a square cylinder found from three-dimensional simulations (left) and two-dimensional simulations (right). Figure from [98].

fluctuations of atmospheric turbulence to aerodynamic forces on the solid. The aerodynamic admittance is used in conjuction with atmospheric turbulence spectra in spectral structural analysis such as the IBDAS model[1]. Typically the aerodynamic admittance is estimated from analytic models [102, 107, 66] or experimentally [55] with the limited available turbulence range of wind tunnels [86, 47]

In the analysis of the aerodynamic admittance [54], the lift coefficient

$$C_L = \frac{L}{\frac{1}{2}\rho U^2 C}, \tag{1.2.1}$$

is assumed to be a linear function of the angle of attack (α), thus

$$C_L(\alpha) = C_{L_0} + C'_L \alpha. \tag{1.2.2}$$

where L denotes the lift force per unit length, C the chord length of the bridge section, ρ the density of the fluid, U the horizontal mean free stream velocity, and $C'_L \equiv \partial C_L / \partial \alpha$. For a fixed rigid structure, the instantaneous angle of attack depends on U and the horizontal and vertical velocity fluctuations u and w, respectively, thus

$$\alpha \approx \frac{w}{U+u}. \tag{1.2.3}$$

By inserting the instantaneous velocity

$$U_i = \sqrt{w^2 + (U+u)^2}, \tag{1.2.4}$$

[1]The IBDAS model is a finite element tool developed by COWI (www.cowi.com)

into Eq. (1.2.1) and combining it with Eqs. (1.2.2) and (1.2.3) the lift force can be expressed as

$$L = \frac{\rho C}{2}\left(C_{L_0}U^2 + 2C_{L_0}Uu + C'_L wU\right), \tag{1.2.5}$$

where terms involving products of fluctuations have been neglected. Thus, the lift force due to the wind fluctuations becomes

$$L_f = \frac{\rho UC}{2}\left[2C_L u + C'_L w\right]. \tag{1.2.6}$$

Assuming that the force due to velocity fluctuations is a stationary random process, the equation can be transformed into the frequency domain [54]:

$$S_{LL} = \left(\frac{\rho UC}{2}\right)^2 \left[4C_L^2 S_{uu}\chi_L^u + C_L'^2 S_{ww}\chi_L^w\right], \tag{1.2.7}$$

where $\chi_L^u = \chi_L^u(\omega)$ and $\chi_L^w = \chi_L^w(\omega)$ denote the admittances due to the spectral density of the horizontal $S_{uu} = S_{uu}(\omega)$ and vertical $S_{ww} = S_{ww}(\omega)$ velocity fluctuations, and ω the angular frequency of the fluctuation. See Appendix. B on analytic and discrete Fourier transforms and the spectral density. It is difficult to distinguish between the contribution from u and w, and instead the lumped aerodynamic admittance is defined:

$$\chi_L(\omega) = \frac{S_{LL}(\omega)}{\left(\frac{1}{2}\rho UC\right)^2\left[4C_L^2 S_{uu}(\omega) + C_L'^2 S_{ww}(\omega)\right]}. \tag{1.2.8}$$

Often C_L is small compared to C'_L and the aerodynamic admittance can be reduced to a relation between the spectral density of the vertical velocity fluctuations S_{ww} and the lift force S_{LL}. That is, fluctuations in the lift force are mainly influenced by the vertical velocity fluctuations. For the pitching moment M the aerodynamic admittance can be derived analogously [54]

$$\chi_M(\omega) = \frac{S_{MM}(\omega)}{\left(\frac{1}{2}\rho UC^2\right)^2\left[4C_M^2 S_{uu}(\omega) + C_M'^2 S_{ww}(\omega)\right]}, \tag{1.2.9}$$

where

$$C_M = \frac{M}{\frac{1}{2}\rho U^2 C^2}, \tag{1.2.10}$$

and $C'_M = \partial C_M/\partial\alpha$. The aerodynamic drag is often an even function of the angle of attack around $\alpha = 0$. Therefore the first order term in the expansion of the drag force vanishes, increasing the dependency on the higher order

terms and thereby restricts a proper definition of the aerodynamic drag admittance.

In spite of the aerodynamic admittance generally being frequency dependent the constant transfer functions

$$S_{LL}(\omega) = \left(\frac{1}{2}\rho C U C'_L\right)^2 S_{ww}(\omega), \tag{1.2.11}$$

$$S_{MM}(\omega) = \left(\frac{1}{2}\rho C^2 U C'_M\right)^2 S_{ww}(\omega), \tag{1.2.12}$$

are often being used, assuming proportionality between the spectrum of the vertical fluctuations S_{ww} and the spectra S_{LL} and S_{MM} by C'_L and C'_M, respectively. This in effect, renders the aerodynamic admittance unity in the whole frequency range. The relation stems from potential theory with the lift force being the superposition of multiple lift force signals, each proportional to a sinusoidal vertical velocity fluctuation.

In the present work the flow past a flat plate and the flow past the Great Belt East bridge is simulated. By sampling time series of the velocity fluctuations, lift forces and pitching moments, the spectra S_{uu}, S_{ww}, S_{LL} and S_{MM} can be computed, and in turn the aerodynamic admittances χ_L and χ_M.

1.3 CFD and fluid-structure interaction

Historically the range of CFD algorithms has grown and the performance and applicability of CFD has improved. The Navier-Stokes equation, governing the flow, may be formulated in a Lagrangian or an Eulerian frame

$$\frac{D\boldsymbol{u}}{Dt} = \frac{\partial \boldsymbol{u}}{\partial t} + \boldsymbol{u}\boldsymbol{\nabla}\cdot\boldsymbol{u} = -\frac{1}{\rho}\boldsymbol{\nabla}p + \nu\boldsymbol{\Delta}\boldsymbol{u}. \tag{1.3.1}$$

Eq. (1.3.1) governs the incompressible velocity field $\boldsymbol{u}$ with density ρ and constant kinematic viscosity ν. For incompressible flow the continuity equation

$$\boldsymbol{\nabla}\cdot\boldsymbol{u} = \boldsymbol{0}. \tag{1.3.2}$$

must be satisfied. Due to the Lagrangian nature of particle methods their use for flow simulation is apparent as the computational elements move like elements of the flow under the influence of the local forcing. Vortex methods are particularly suitable for exterior flow simulation as vorticity is confined

to the wake and boundary layer regions. The vorticity $\boldsymbol{\omega} = \boldsymbol{\nabla} \times \boldsymbol{u}$ and position $\boldsymbol{x}$ of vortex particles is governed by the vorticity equation

$$\frac{D\boldsymbol{\omega}}{Dt} = \boldsymbol{\omega} \cdot \boldsymbol{\nabla}\boldsymbol{u} + \nu\boldsymbol{\nabla}^2\boldsymbol{\omega} \tag{1.3.3}$$

$$\frac{d\boldsymbol{x}}{dt} = \boldsymbol{u} \tag{1.3.4}$$

The velocity $\boldsymbol{u} = \boldsymbol{\nabla} \times \boldsymbol{\Psi} + \boldsymbol{U}_\infty$ is defined as the sum of curl of the stream function $\boldsymbol{\Psi}$ and an irrotational free-stream velocity $\boldsymbol{U}_\infty$. The stream function is found by solving the Poisson equations

$$\Delta\boldsymbol{\Psi} = -\boldsymbol{\omega} \tag{1.3.5}$$

which is derived from the definition of the vorticity and velocity. Neither Dirichlet nor Neumann domain boundary condition need to be forced as vortex methods offer unbounded solutions where the velocity goes to asymptotically to a free stream velocity at infinity. Using boundary element methods [113] the no-slip condition can be imposed at the solid boundaries and hence alleviates the time consuming mesh generation required in Eulerian methods. Furthermore, the limitations to the discretisation of the time integration of particle methods are less constraining than those of grid based Eulerian methods [25].

Rosenhead [87] was the first to compute, by hand, the motion of a shear layer numerically, approximating it by discrete vortex particles, see Fig 1.4. Since Rosenhead's numerical analysis in 1931 vortex methods have been improved and are widely used in academia and by the industry for flow modelling [97, 52, 21, 64, 77, 116, 9, 101, 38].

Vortex particles with finite core [13] ensure stable and bounded motion of the continuous vorticity field. Through remeshing of the particles [68, 51] and the entailing regular placement, spatial convergence is ensured.

1.3.1 Computational scaling

To advance the vortex particles as part of the Lagrangian time integration the velocity induced by the vortex particles must be determined. This may become computationally expensive. As the number of computational elements, N, increases the computational work scales disproportionally ($\mathcal{O}(N^2)$) as all particles interact with each other; in effect preventing large simulations. Two families of algorithms have been developed to improve the computational scaling.

The fast multipole method (FMM) [3, 40, 8, 12] groups particles through recursive multipole expansions and evaluates the far-field interaction through

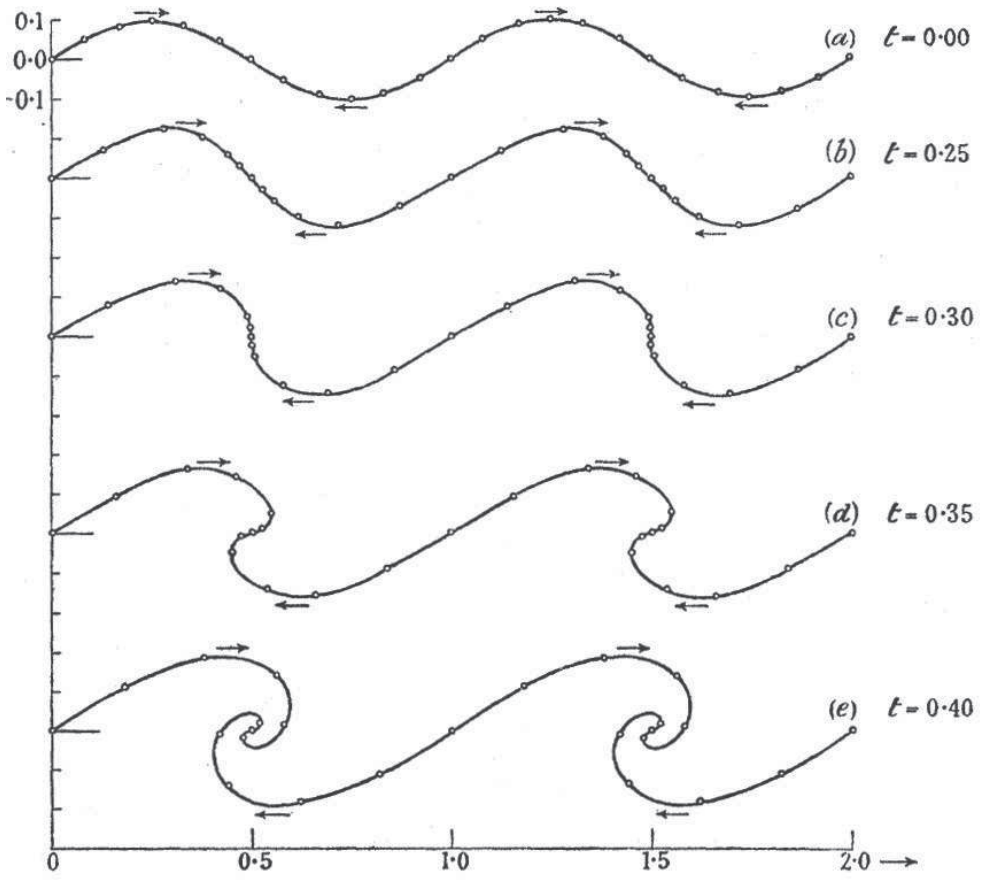


Figure 1.4: The evolution of a shear layer computed, by hand, by Rosenhead in 1931. Figure from [87].

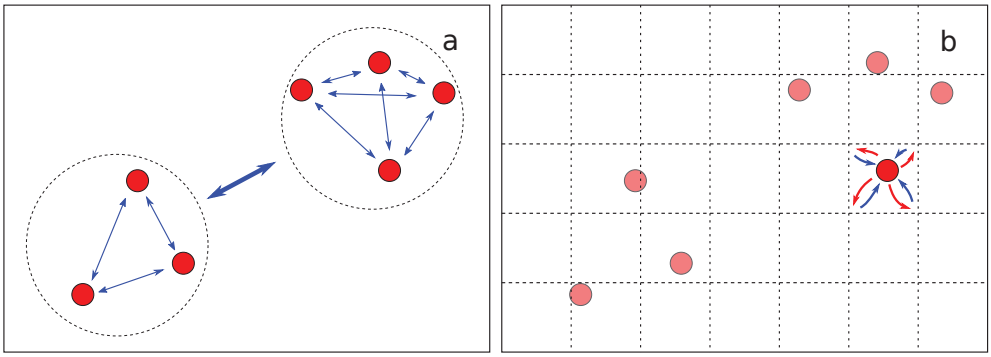


Figure 1.5: Algorithms to compute the velocity (blue) induced by the vortex particles (red circles). (a) The FMM groups particles and computes collective velocity contributions. Within groups all particles interact. (b) The VIC algorithm interpolates the vorticity of each particle to an underlying mesh (dashed lines), computes the velocity on the mesh, and interpolates the velocity back to the particle.

recursive Taylor expansions resulting in linear scaling ($\mathcal{O}(N)$). This is outlined in Fig. 1.5a. In spite of the optimal $\mathcal{O}(N)$ scaling a big prefactor makes parallel three-dimensional implementations of the FMM less efficient than the other alternative [92]: The hybrid vortex-in-cell algorithm (VIC). [7, 17].

The VIC interpolates the particle strength to an underlying mesh. Fast Fourier transforms (FFT) are used to calculate the stream function on the mesh, from which the velocity can be computed. This is outlined in Fig. 1.5b. This yields $\mathcal{O}(N \log N)$ computational scaling. The mesh representation of the vorticity field may be used to efficiently calculate the Navier-Stokes equation right hand side. Subsequently the time rate change of vortex particle strength and position is interpolated back to the particles and integrated in accordance with the Lagrangian Navier-Stokes equation. The drawback of the Cartesian uniform meshes, required by the FFTs, is the lack of mesh adaptivity. If required, particles may be distributed more unevenly but the accuracy of the velocity computation will be dictated by the uniform mesh resolution.

Non-uniform resolution in particle methods has been achieved with conform mappings [77, 24], coupling differently resolved domains through Dirichlet boundary conditions [99, 34] and FMM multilevel particle remeshing algorithms [6]. The latter reference, which introduces overlaid meshes and overlapping buffers, serves as the starting point for the present multiresolution mesh based velocity algorithm. In the present work meshes of varying

resolution are staggered and the buffers are adapted accordingly, thereby alleviating the discrete sub-sectioning of the domain. The local refinement is determined a priori in the current work, but the presented algorithm does not prevent the formation or repositioning of the refined areas during the simulation.

Due to the periodicity of the FFT, VIC simulations have been constrained to periodic boundary conditions. However, through modification of the vorticity field, non-periodic [45, 46, 69] or semi-periodic [10] flows can be simulated in spite of the periodic nature of the FFT algorithm.

The need to do parallel computations arises as computational requirements grow with the increasingly large, three-dimensional simulations of more complex time dependent fluid flow. Parallelisation may be aided by software libraries such as the parallel particle mesh (PPM) library [92] to ease tedious and complex program structural procedures. The VIC algorithm has previously been implemented in parallel to simulate large flow problems, e.g. trailing vortices with billions of particles [9]. Another efficient hybrid scheme [22, 72, 111, 19] decomposes space into smaller VIC problems with boundary conditions provided by an FMM.

In vortex methods the velocity field is found as a the curl of a stream function. The stream function corresponds to the vector potential of electro magnetics and is found from the Poisson equation as the gravitational potential is. Therefore parts of the present work are relevant for physics simulations governed by the Poisson equation and physics depending on Green's function solutions. This could be celestial mechanics [36], electro dynamics or molecular dynamics.

1.3.2 Solid boundaries

As mentioned, one of the strengths of particle methods is their mesh-free nature. Thus the complex process of creating meshes conforming with solid surfaces is not required in order to apply the no-slip boundary condition. Instead immersed boundary methods [76, 78] and boundary element methods [113] enforce the correct boundaries only requiring a discrete representation of the surface. These methods involve solving a system of linear equations for the strength of the surface-bound vorticity. However, for highly resolved surface discretisations the full Gaussian elimination of the system becomes very costly. Iterative methods can be applied to solve the linear system of equations but require computing the velocity field every iteration.

Alternatively solid surfaces can be enforced by Brinkman penalization as

an additional term to the right hand side of the governing equation

$$\frac{D\boldsymbol{\omega}}{Dt} = \boldsymbol{\omega} \cdot \boldsymbol{\nabla}\boldsymbol{u} + \nu\boldsymbol{\nabla}^2\boldsymbol{\omega} + \lambda\boldsymbol{\nabla} \times [\chi(\boldsymbol{u}_s - \boldsymbol{u})] \,, \tag{1.3.6}$$

where λ is a numerical parameter corresponding to an inverse porosity, χ defines the solid domain and $\boldsymbol{u}_s$ the velocity field of the solid. Initially used to apply volume forcing to model the flow in porous media [2] Brinkman penalization has been used 'in the limit' of zero porosity to model solid boundaries by penalizing the velocity flow at the immersed surface [73, 39]. At high Reynolds numbers it is necessary to penalize the entire solid volume [90]. A consequence of the Brinkman penalization is the introduction of the inverse porosity as a numerical parameter, the penalization parameter. Angot showed [1] that the penalized velocity converges to the solution of the Navier-Stokes equation with the correct solid boundary conditions. For the flow to remain numerically stable the penalization parameter is constrained by a factor of the inverse time step size.

The use of Brinkman penalization has subsequently been used in vortex methods [50] and the smooth solid mask has been introduced to accommodate the Fourier transform [39, 82, 89]. The penalization can be unconditionally stable by imposing it in an implicit first order time integration split-step [20]. In the present work it is shown that the implicit penalization corresponds to an interpolation between the fluid and solid velocity fields.

1.3.3 Implementations

The numerical simulations of the work presented here has been carried out using four different vortex particle implementations:

`DVMFLOW`	Two-dimensional discrete vortex method [108] used in academics and the industry [63].
`vic20`	Multiresolution vortex-in-cell [83] implementation in `MATLAB`.
`wvic`	Periodic vortex-in-cell [9] parallel implementation based on PPM.
`naga`	Free-space and periodic vortex-in-cell parallel implementation based on PPM.

`naga` has been implemented with the aim and structure to do three-dimensional multiresolution VIC simulations. The required modifications to PPM have not been completed and therefore three-dimensional multiresolution is not presented as part of this work. However, algorithmic considerations regarding the PPM implementation are presented in chapter 4.

Chapter 2

Admittance

During the design phase of large span bridges the aerodynamic admittance is often determined experimentally. This is complicated by the limited range of turbulence length scales that can be generated in wind tunnels [86, 47] and the accuracy of the measured aerodynamic forces. The discrete vortex model (DVM) has been used extensively in the industry to determine aerodynamic force coefficients for bridge decks. The DVM implementation `DVMFLOW` is used by the consulting company COWI[1] to simulate the flow past bridge sections [63, 58, 106, 62]. It is in this framework the following work has been implemented.

So far these studies have been limited to laminar flow simulations and only a few studies have considered the modelling of turbulence using two-dimensional particle vortex methods cf. [74, 79]. In the recent work of Prendergast and McRobie [80, 79], oncoming turbulence was modelled by seeding the free stream with vortex particles and simulations were performed to study buffeting in bridge aerodynamics. The present work extends the model of Prendergast and McRobie to enable, for the first time, simulations of the aerodynamic admittance in bluff body aerodynamics.

This work has previously been publish in *Journal of Wind Engineering & Industrial Aerodynamics* [84].

2.1 The discrete vortex method

An incompressible flow with constant kinematic viscosity ν is governed by the 2D Navier-Stokes equations in vorticity form

$$\frac{D\omega_u}{Dt} = \frac{\partial \omega_u}{\partial t} + \boldsymbol{u} \cdot \boldsymbol{\nabla}\omega_u = \nu \Delta \omega_u. \tag{2.1.1}$$

[1] www.cowi.com

The fluid velocity $\boldsymbol{u}$ is computed from the free stream vorticity $\boldsymbol{\omega}_u = \omega_u \boldsymbol{e}_z$ and the stream function ψ:

$$\boldsymbol{\omega}_u = \boldsymbol{\nabla} \times \boldsymbol{u}, \tag{2.1.2}$$

$$\boldsymbol{u} = \boldsymbol{U}_\infty + \boldsymbol{\nabla} \times (\psi \boldsymbol{e}_z). \tag{2.1.3}$$

$\boldsymbol{U}_\infty$ is the free-stream velocity. Combining Eqs. (2.1.2) and (2.1.3) and assuming ψ to be divergence free leads to the Poisson equation

$$\Delta \psi = -\omega_u. \tag{2.1.4}$$

Equation (2.1.4) forms the basis of hybrid vortex particle-mesh methods such as the Vortex-In-Cell algorithm [7, 92, 69, 9]. In the following, the Poisson equation is solved using a Green's function solution:

$$\psi(\boldsymbol{x}) = \Psi + \int G(\boldsymbol{x} - \boldsymbol{y}) \omega_u(\boldsymbol{y}) \, \boldsymbol{dy}, \tag{2.1.5}$$

$$\boldsymbol{u}(\boldsymbol{x}) = \boldsymbol{U}_\infty - \int \boldsymbol{K}(\boldsymbol{x} - \boldsymbol{y}) \omega_u(\boldsymbol{y}) \, \boldsymbol{dy}, \tag{2.1.6}$$

where Ψ is the far-field stream function, and G and $\boldsymbol{K}$ the corresponding 2D Green's functions:

$$G = -\frac{1}{2\pi} \log |x|, \tag{2.1.7}$$

$$\boldsymbol{K} = \frac{1}{2\pi} \frac{\boldsymbol{x}}{|\boldsymbol{x}|^2} \times . \tag{2.1.8}$$

The vorticity field is approximated by discrete vortex particles carrying circulation:

$$\Gamma = \int_{\mathcal{A}} \omega_u dA = \int_{\mathcal{S}} \boldsymbol{u} \cdot \boldsymbol{ds}, \tag{2.1.9}$$

where $\mathcal{A}$ is the area of the particle bounded by $\mathcal{S}$. The vorticity field is discretised by a superposition of N discrete vortex particles:

$$\omega_u^\epsilon(\boldsymbol{x}) = \sum_i^N \zeta_\epsilon(\boldsymbol{x}_i - \boldsymbol{x}) \, \Gamma_i, \tag{2.1.10}$$

where $\zeta_\epsilon(\boldsymbol{x})$ is a smooth approximate to the Dirac delta function: $\zeta_\epsilon(\boldsymbol{x}) = \frac{1}{\epsilon^2}\zeta(|\boldsymbol{x}|/\epsilon)$, and ϵ is the smoothing radius. The present study uses the second order Gaussian kernel: $\zeta(r) = \frac{1}{2\pi} e^{-r^2/2}$ cf. e.g. [112].

The discrete kinematic relation governing the flow is obtained from Eqs. (2.1.6) and (2.1.10):

$$\boldsymbol{u}(\boldsymbol{x}) = \boldsymbol{U}_\infty - \sum_i^N \boldsymbol{K}_\epsilon(\boldsymbol{x} - \boldsymbol{x}_i) \times \Gamma_i \boldsymbol{e}_z, \tag{2.1.11}$$

where $\boldsymbol{K}_\epsilon = -(q_\epsilon(\boldsymbol{x})/|\boldsymbol{x}|^2)\boldsymbol{x}$ is the smooth velocity kernel, and $q_\epsilon(\boldsymbol{x}) = q(|\boldsymbol{x}|/\epsilon)$, and $q(r) = \frac{1}{2\pi}(1 - e^{-r^2/2})$.

The motion and strength of the discrete particles is solved using viscous splitting [13]. Hence, the particles are first convected:

$$\frac{d\boldsymbol{x}_p}{dt} = \boldsymbol{u}(\boldsymbol{x}_p), \tag{2.1.12}$$

$$\frac{d\boldsymbol{\omega}_u}{dt} = 0, \tag{2.1.13}$$

and subsequently diffused:

$$\frac{d\boldsymbol{x}_p}{dt} = 0, \tag{2.1.14}$$

$$\frac{d\boldsymbol{\omega}_u}{dt} = \nu\Delta\boldsymbol{\omega}_u. \tag{2.1.15}$$

In the present `DVMFLOW` implementation, the convection step is solved using first or second order explicit time integration schemes, and diffusion is modelled using the method of random walks [13]. Hence, after the convection step (Eq. (2.1.12)), the position of the vortex particles is perturbed with a random displacement, drawn from a normal distribution with zero mean and variance $2\nu\Delta t$. Δt denotes the simulation time step.

The solid boundaries are discretised using a boundary element technique [113, 114]. The no-penetration condition is enforced by determining the strength of the vortex sheets (γ) on the panels, which forms a linear system of equations. A unique solution is obtained by imposing Kelvin's circulation theorem [108]:

$$\sum \Gamma = 0. \tag{2.1.16}$$

The introduction of upstream vortex particles with a total non-zero circulation Γ_T requires a modification of Eq. (2.1.16) such that $\sum \Gamma = \sum \Gamma_T$.

The computational efficiency of the discrete vortex method is closely related to the solution of the N-body problem implied by Eqs. (2.1.11) and (2.1.12) which nominally scales as $O(N^2)$. The `DVMFLOW` implementation uses the Fast Multipole Method [40, 8] to achieve an optimal $O(N)$ scaling.

The forces and moments are calculated from the pressure distribution which in turn is calculated from the distribution of the vortex sheet along the boundary s:

$$\frac{1}{\rho}\frac{\partial p}{\partial s} = -\frac{\partial \gamma}{\partial t}. \tag{2.1.17}$$

The derivatives C'_L and C'_M are found by finite differences of C_L and C_M measured at different angles of attack assuming either a laminar or turbulent oncoming flow. Laminar flow simulations are used similar to the practice of wind tunnel testing.

2.2 Synthesizing turbulence

To introduce turbulence into the oncoming flow of a simulation a time series of vortex particles is generated prior to the simulation [80]. These particles are inserted into the flow during the simulation at a fixed insert rate. The vortex particles are convected downstream forming a band of particles which induce turbulent velocity fluctuations. Initially turbulent velocity series are generated on the nodes of a regular vertical grid using the Shinozuka-Deodatis method [95, 32], and in accordance with the spectral densities of the atmospheric velocity fluctuations. The grid consists of N_p vertically aligned quadratic cells, see Fig. 2.1. In each point two velocity time series are generated: one in the streamwise (horizontal) direction and one in crosswise (vertical) direction, u and w respectively. Presently these velocity time series are based on the modified von Kármán spectra of the Engineering Sciences Data Unit (ESDU) [103]

$$\begin{aligned}\frac{nS_{uu}}{\sigma_u^2} &= \beta_1 \frac{2.987 n_u/\alpha}{\left[1+(2\pi n_u/\alpha)^2\right]^{\frac{5}{6}}} \\ &\quad + \beta_2 \frac{1.294 n_u/\alpha}{\left[1+(\pi n_u/\alpha)^2\right]^{\frac{5}{6}}} F_1, \qquad (2.2.1)\\ \frac{nS_{ww}}{\sigma_w^2} &= \beta_1 \frac{2.987\left[1+(8/3)(4\pi n_w/\alpha)^2\right](n_w/\alpha)}{\left[1+(4\pi n_w/\alpha)^2\right]^{\frac{11}{6}}} \\ &\quad + \beta_2 \frac{1.294 n_w/\alpha}{\left[1+(2\pi n_w/\alpha)^2\right]^{\frac{5}{6}}} F_2, \qquad (2.2.2)\end{aligned}$$

see Appendix A for a description of the involved terms. Atmospheric turbulence is anisotropic, thus two spectra are needed: one for the horizontal

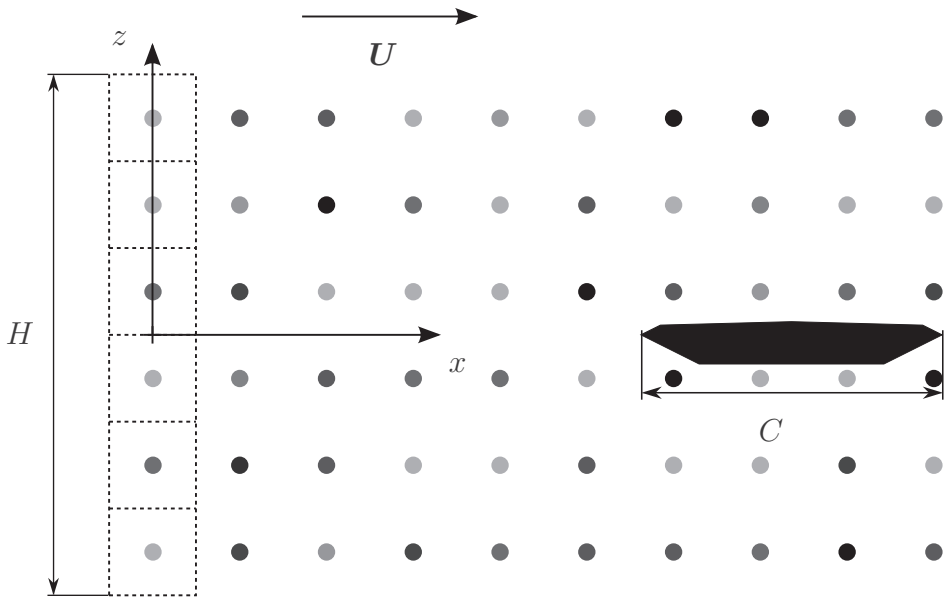


Figure 2.1: From the grid at the left vortex particles are released and convected downstream by the flow induced by the vortex particles of the flow and the free stream velocity U. Coordinates (x,z) are given relative to the center of the H high grid, (u,w) are the respective velocity components. Solid objects with chord length C inserted in the flow are placed on the x-axis.

and one for the vertical fluctuations. The magnitude and distribution of the spectra depend on various physical parameters such as the surface roughness length scale (z_0), the height above ground level (h), and the magnitude of the free stream velocity (U). However the available parameters can only be varied within a certain range and with limited influence on the turbulence intensity

$$\begin{aligned} I_u &= \frac{\sigma_u}{U}, \\ I_w &= \frac{\sigma_w}{U}. \end{aligned} \tag{2.2.3}$$

Hence, varying the physical parameters mostly influences the magnitude of the atmospheric velocity spectral density while the distribution is largely unaffected cf. [18, 37]. Therefore to obtain a specific turbulence intensity $\tilde{I}_w = \tilde{\sigma}_w/U$ the analytic spectral density $S_{ww}(\omega)$ is scaled

$$\tilde{S}_{ww}(\omega) = \left(\frac{\sigma_w}{\tilde{\sigma}_w}\right)^2 S_{ww}(\omega), \tag{2.2.4}$$

to match the target intensity:

$$\int \tilde{S}_{ww}(\omega)d\omega = \tilde{\sigma}_w^2. \tag{2.2.5}$$

The spectral densities are discretised in N_f discrete frequencies

$$\omega_k = k\Delta\omega, \quad k = 0, 1, 2, ..., N_f - 1, \tag{2.2.6}$$

with a uniform spacing

$$\Delta\omega = \frac{\omega_{\text{up}}}{N_f}. \tag{2.2.7}$$

The upper cut-off frequency ω_{up} is chosen such that the energy of the discarded frequency range is negligible. Furthermore the Nyquist criterion

$$\omega_{\text{up}} \leq \frac{\pi}{\Delta t_{\text{gen}}}, \tag{2.2.8}$$

is fulfilled to avoid aliasing. Δt_{gen} is the time step at which velocities are generated on the grid, and $1/\Delta t_{\text{gen}}$ is the corresponding insert rate.

The velocity series in points m and n on the grid are spatially correlated through the coherence function [30, 88]

$$\text{Coh}_{u_m u_n}(\omega) = e^{-\hat{f}}, \tag{2.2.9}$$

with the decay function

$$\hat{f} = \frac{\omega}{2\pi} \frac{\sqrt{C_{ux}^2 (x_m - x_n)^2 + C_{uz}^2 (z_m - z_n)^2}}{0.5\,(U(z_m) + U(z_n))}. \tag{2.2.10}$$

Where $C_{ux} = 3$ and $C_{uz} = 10$ are decay coefficients [96]. The spatial correlation for the vertical velocities is analogous to Eqs. (2.2.9) and (2.2.10), and the decay coefficients are assumed $C_{wx} = C_{ux}$, $C_{wz} = C_{uz}$ [79]. Hereby the influence of a velocity series in m to a series in n becomes

$$S_{u_m u_n}(\omega) = \sqrt{S_{u_m u_m}(\omega) S_{u_n u_n}(\omega)} \mathrm{Coh}_{u_m u_n}(\omega) e^{i\theta_{u_m u_n}(\omega)}. \tag{2.2.11}$$

The angular phase shift

$$\theta_{u_m u_n}(\omega) = \omega \frac{x_n - x_m}{U}, \tag{2.2.12}$$

is defined in correspondence with Taylor's Hypothesis, i.e. the phase shift is equal to the duration for the flow to move between the points m and n times the frequency. For $m = n$ it is seen that $S_{u_m u_n}$ is given directly from S_{uu}. It is assumed that there is only spatial correlation between velocity components in the same direction [79], i.e. that there is no correlation between the u and w velocities. The contribution of all velocities to each other is contained in the cross spectral matrix:

$$\boldsymbol{S}(\omega) = \begin{bmatrix} [S_{u_m u_n} \;\; m, n \leq N_p] & \mathbf{0} \\ \mathbf{0} & [S_{w_m w_n} \;\; m, n \leq N_p] \end{bmatrix}, \tag{2.2.13}$$

Let m be the index of a velocity process in a point and by velocity process understand a velocity series either horizontal or vertical. It follows that there will be twice the number of velocity processes as the number of points in the grid N_p. Then

$$\begin{aligned} f_m(t) &= \sqrt{2\Delta\omega} \sum_{n=1}^{2N_g} \sum_{k=1}^{N_f - 1} |H_{mn}(\omega_k)| \cos \beta(t), \\ \beta(t) &= \omega_k t + \theta_{mn}(\omega_k) + \phi_{nk}, \end{aligned} \tag{2.2.14}$$

is the mth velocity process [95] found by summing a series of N_f cosine waves from each of the $2N_p$ velocity processes, including m. In effect the summation over m is only to N_p with no correlation between vertical and horizontal velocities in the cross spectral matrix $\boldsymbol{S}(\omega_k)$ cf. Eq. (2.2.13). The amplitudes

of the cosine waves are determined from the Cholesky decomposition $\boldsymbol{H}(\omega_k)$ of $\boldsymbol{S}(\omega_k)$. Furthermore

$$\theta_{mn}(\omega_k) = \arctan\left(\frac{\mathrm{Im}[H_{mn}(\omega_k)]}{\mathrm{Re}[H_{mn}(\omega_k)]}\right), \tag{2.2.15}$$

is the complex argument of $H_{mn}(\omega_k)$ and ϕ_{nk} is a random phase in the interval $[0; 2\pi]$.

The efficiency of the process is significantly improved by using Fast Fourier Transforms (FFTs) to carry out the summation [32]. The velocity series becomes

$$f_m(t_r) = \sqrt{2\Delta\omega}\sum_{n=1}^{N_p}\mathrm{Re}\left[C_{mnr}\right], \tag{2.2.16}$$

where

$$C_{mnr} = \sum_{k=0}^{2N_f-1} c_{mnk}e^{i\frac{2\pi}{2N_f}kr}, \tag{2.2.17}$$

is the Fourier transform of

$$c_{mnk} = \left|H_{mn}(\omega_k)\right| e^{i(\theta_{mn}(\omega_k)+\phi_{nk})}. \tag{2.2.18}$$

Using the maximum discrete time step that fulfils the Nyquist criterion Eq. (2.2.8), the discrete time becomes

$$t_r = r\Delta t_{\mathrm{gen}}, \qquad r \in [0; 2N_f - 1], \tag{2.2.19}$$

For each of the quadratic cells of the grid the circulation is integrated from the grid node velocities using the trapezoidal rule [79]. The circulation is corrected in magnitude by a factor K such that

$$\Gamma = K\Gamma_{\mathrm{trapezoidal}}. \tag{2.2.20}$$

This accounts for the mismatch between the linear assumption of the integration and the actual circular velocity contours of a point vortex. The actual distribution of circulation from a singular point vortex of strength Γ_{vortex} on a cell with side length Δx is not linear but given by

$$\frac{\mathrm{d}\Gamma}{\mathrm{d}s} = \frac{\Gamma_{\mathrm{vortex}}}{\pi\Delta x(s^2+1)}, s \in [-1; 1], \tag{2.2.21}$$

see Fig. 2.2. By integrating Eq. (2.2.21) around the cell boundary it is

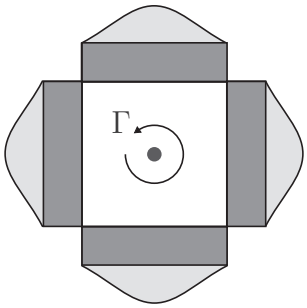


Figure 2.2: The circulation from a point vortex integrated with the corner point approximation on a cell with side length Δx, is marked by the dark gray area. The actual non-linear distribution $\frac{\mathrm{d}\Gamma}{\mathrm{d}s} = \frac{\Gamma_{\mathrm{vortex}}}{\pi\Delta x(s^2+1)}, s \in [-1;1]$ is the unification of the dark and the light gray areas.

seen that $\Gamma = \Gamma_{\mathrm{vortex}}$ if $K = \pi/2$. For each cell a particle is associated with the corresponding circulation and inserted during the flow simulation at every $\Delta_p = \Delta t_{\mathrm{gen}}/\Delta t$ time step. The side length of the quadratic cells $\Delta x = \Delta_p \Delta t U$ corresponds to the distance the particles are convected by the free stream velocity between subsequent particle releases. As the particles are being convected downstream they form a band of particles with a total non-zero circulation $\sum \Gamma_T$. The final condition for the panel strength of the boundary elements Eq. (2.1.16) is modified such that $\sum \Gamma = \sum \Gamma_T$. Imposing total zero circulation would otherwise affect the panel strengths and render force calculation by surface pressure distribution impossible. Alternatively the particle strengths can be modified by subtracting the mean particle turbulence circulation from the particles being released such that no net circulation is added. However, this disrupts the streamwise correlation between the particles and leads to strongly reduced energies in the low frequency region of spectral density of the resulting simulation velocities.

2.3 Validation, numerical parameters and results

Simulations of the turbulent flow past a flat plate and past the Great Belt East bridge are performed to test and validate the current implementation. The admittance for the flow past the flat plate serves as a reference case and allow systematic variation of the key numerical parameters. The reference parameters related to the discretisation of the spectral density of atmospheric

turbulence is the upper angular frequency $\omega_{\mathrm{u}} = 36.7\,\mathrm{rad/s}$ and the number of discrete frequencies $N_f = 4096$. When integrating the grid velocities to particle strength a correction factor $K = \pi/2$ is used. The variation of the spectra with respect to the point of sampling is examined, and the dependency of the number of particles N_p per release as well as the particle insert interval Δ_p is investigated. Reference values for these parameters are $N_p = 120$ and $\Delta_p = 4$.

The atmospheric turbulence is reconstructed from the ESDU spectra, defined in Appendix A using the reference parameters. The reference spectra is subsequently scaled to meet a specified vertical turbulence intensity of $I_w = 5\,\%$.

Introducing a solid structure into the flow also introduces an additional length scale, i.e. the chord length (C), and hence the Reynolds number $\mathrm{Re} = UC/\nu$. This prompts studying the influence of Re and turbulence intensity I_w on the aerodynamic admittance. As reference the flow is simulated at Re = 10.000 to ensure a relatively thin boundary layer and to allow comparison with the potential flow solution. For the boundary layer to remain stable [108] and to ensure sufficient spatial resolution the simulations are not carried out at higher Reynolds numbers. The relevant Reynolds number for full scale bridges is $\mathcal{O}(10^8)$ and for the wind tunnel tests typically $\mathcal{O}(10^5)$. However, to allow comparison with the results obtained for the flat plate Re = 10.000 is maintained for the bridge simulations.

The velocity time series and particle strengths are generated in SI-units and non-dimensionalised before being used as input for the `dvmflow` simulations. As characteristic length a typical value for the bridge chord length $C = 30\,\mathrm{m}$ is used, for both bridge section and the flat plate benchmark. The characteristic free stream velocity is $U = 35\,\mathrm{m/s}$ which is a typical design wind speed in bridge engineering. The recorded velocities, forces and pitching moments are re-dimensionalised and analysed. As the aim of this text is to investigate flow properties of solid objects all positions have been given in units of C. For comparison purposes this is also the case when investigating the flow when no solid body is present.

Spectral densities are statistical properties of a sample or a signal and often contain a significant amount of noise. To reduce noise the signals have been subsampled with 50 % overlap and the resulting spectra averaged. This significantly improves the consistency of the spectra but at the cost of reduced low-frequency information. Further noise reduction can be achieved by applying window functions at the cost of uncertainty in the magnitude of the spectra [41]. Window functions preserve the shape of the spectrum and reduce noise, thus admittances have been calculated from spectra based on windowed samples, as any change of the magnitude cancels out. Before

sampling any velocity and force the simulation is carried out until both the sampling points and the structures are well immersed in the turbulent flow. Most signals in the present work consist of 38.000 samples which are divided into 2 to 10 subsamples. Though some uncertainty remains, this reduces noise significantly at the cost of the frequency range. To be able to compare similar spectra in the same figure, approximations by N-point Bézier curves have been used to remove noise when necessary. N-point Bézier curves use all N samples for the approximation. The typical range of interest for structural analysis of long span bridges is 0.05–1.0 Hz. For pedestrian bridges the range of interest is typically in the range of 0.5–5.0 Hz. In angular frequency these intervals are 0.31–6.3 rad/s and 3.14–31.4 rad/s respectively. In the following the terms: low-, mid- and high-range frequencies refer approximately to the frequency intervals separated by $\omega = 0.8\,\mathrm{rad/s}$ and $\omega = 10\,\mathrm{rad/s}$.

2.4 Validation of power spectral density of the velocity

When generating the upstream particles many parameters influence the simulated turbulent velocities. The optimal value of the numerical parameters are not necessarily obvious, nor can they be chosen freely due to computational constraints. These parameters are investigated in this section.

2.4.1 Sensitivity to upper cutoff and spectral resolution

The upper cutoff of the frequency (ω_u) and the finite number of discrete frequencies (N_f) at which the velocity spectrum is defined will affect the resulting velocity signal. Both the velocity signal on the grid and in turn the vortex induced velocities in the simulation are affected. These parameters are studied in the following and the results are compared to theoretical values.

The upper cutoff (ω_u) is chosen above the frequency range of interest such that the turbulent energies σ_u^2 and σ_w^2 are conserved to the extent possible. Fig. 2.3(a) shows the discarded energy

$$\eta(\omega_\mathrm{u}) = \left(1 - \frac{1}{\sigma^2}\int_0^{\omega_\mathrm{u}} S\mathrm{d}\omega\right) \cdot 100\,\%, \tag{2.4.1}$$

as a function of ω_u. Thus, the reference cutoff $\omega_\mathrm{u} = 36.7\,\mathrm{rad/s}$, entails a loss of horizontal turbulent energy of $\eta_u = 1.7\,\%$ and $\eta_w = 6.7\,\%$ vertical velocity

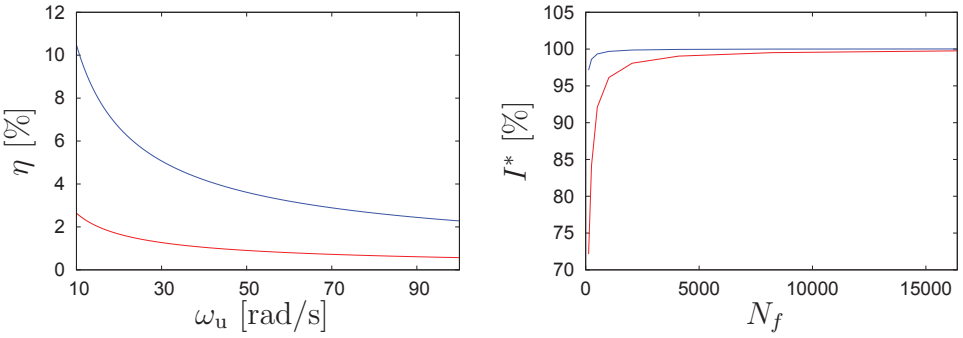


Figure 2.3: (a) Deficiency of energy $\eta = \left(1 - \int_0^{\omega_u} S d\omega / \sigma^2\right) \cdot 100\,\%$ as a function of upper cutoff ω_u for the spectral density of the horizontal (red) and vertical (blue) velocities. (b) Normalized turbulence intensity $I^* = \frac{I_{RMS}}{I_{target}} \cdot 100\,\%$ for the horizontal (red) and vertical (blue) velocity time series on the grid as a function of the number of discrete frequencies N_f. The energy loss due to ω_u has been accounted for in I_{target}.

energy. For the vertical velocity energy loss to be reduced to 1 % the upper cutoff must be approximately 20 times higher. Thus for a constant $\Delta\omega$ the complete energy of the spectrum cannot be expected to be preserved as the memory requirements scale as $\mathcal{O}(N_f N_p^2)$.

To conserve the energy of the spectra for $\omega \leq \omega_u$ a sufficient number of discrete frequencies (N_f) is required. Fig. 2.3(b) shows the turbulence intensities I_u and I_w of the grid velocity series as a function of N_f. The turbulence intensities have been normalized by $I_u\sqrt{1-\eta_u}$, the expected values corrected by the effect of the upper cutoff. The vertical turbulence intensity I_w converges faster than I_u and for $N_f = 4096$ they are 99.96 % and 99.04 % of the expected values respectively. The energy loss from the finite frequency range is more significant than that caused by the resolution of the spectrum. In addition it has been found that the grid velocities are in good agreement with the prescribed spectra, both in terms of the turbulent energy σ_u^2 and σ_w^2 and also the spectral distribution: S_{uu} and S_{ww}, see Fig. 2.4.

2.4.2 Effects of frequency discretisation and circulation integration

While the energy of the time series of the grid velocity is a first indicator of the validity of the synthesis, the spectral density of the velocity time series sampled during the simulation shows more detail of the method, cf. Fig. 2.5. In the following the effect of the discrete frequencies (N_f) on the velocity

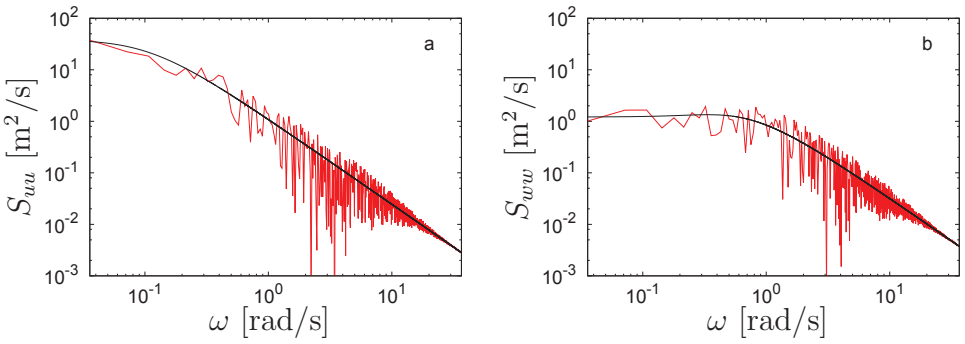


Figure 2.4: Spectral density of (a) horizontal and (b) vertical velocity velocity grid signal.

spectra sampled during the simulation is studied.

Both the horizontal (S_{uu}) and vertical (S_{ww}) spectra show a significant dependency on the number of discrete frequencies N_f, see Figs. 2.5(a) and 2.5(b) respectively. For both the horizontal and vertical spectra, the high frequency range has converged at just 128 discrete frequencies. The higher the spectral resolution the further the spectra converge into the low frequency range and $N_f \geq 2048$ the spectra have converged for all but the very lowest frequencies. The atmospheric turbulence spectra have been discretised with the lowest non-zero discrete frequency $\Delta\omega$. $\Delta\omega$ is indicated for each spectral resolution by a vertical line coloured similarly to the corresponding spectrum, and below $\Delta\omega$ the sampled spectra contain little energy. For the coarse discretization ($N_f = 128$) the discrete frequencies are clearly visible, indicating that exactly the frequencies of interest are preserved by the conversion from grid velocities to particle induced velocities. $N_f = 4096$ is chosen as a compromise between accuracy in the low frequency range and the required memory and computational resources.

The spectral densities of the time series of the grid velocity show perfect match to the target spectra, Fig. 2.4, whereas the spectral density of the simulation deviate from the target. Hence the deviations are a result of the method of converting grid velocity to particle strengths and the finite release rate of the upstream particles. The strongly deviating low frequency range energy of S_{ww} is observed only for the frequencies at which the atmospheric turbulence spectrum has been discretised, Fig. 2.5. This indicates that the deviation is primarily an effect of converting grid velocities to particle strengths and not an effect of the actual release of particles during the simulation. The latter is investigated in section 2.4.6. It is worth noting how well the values of the discrete frequencies are preserved in the process of

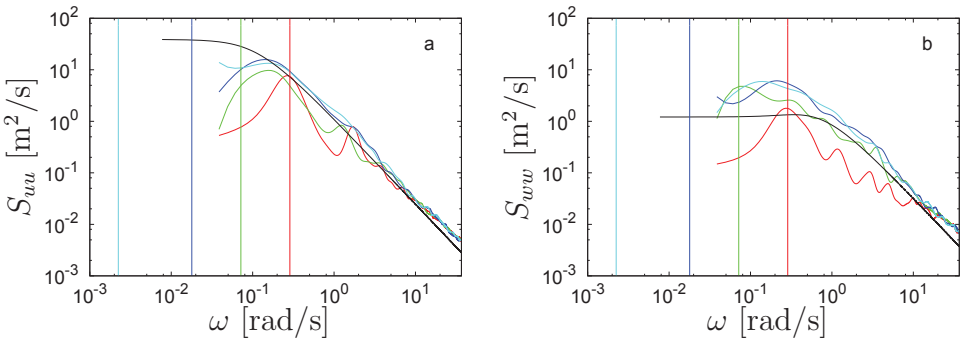


Figure 2.5: The spectral density (a) S_{uu} and (b) S_{ww} of the velocity from the simulations, using turbulence series based on analytic spectra with varying number of discrete frequencies N_f: 128 (red), 512 (green), 2048 (blue), and 16384 (cyan). The target spectrum (black) is shown as reference. The highest frequency in the range is ω_{u}. The vertical lines indicate the respective $\Delta\omega$. The spectra are averages from 10 subsamples and have been smoothened by N-point Béziers.

converting grid velocities to particle strengths. S_{ww} is above target for the whole frequency range, see Fig. 2.5(b), as is also the case for S_{uu} except for the low frequency region in which S_{uu} is weaker than target, see Fig. 2.5(a).

2.4.3 Influence of circulation correction

The correction factor, $K = \pi/2$ cf. Eq. (2.2.20), is based on the simple comparison of the integral over the grid velocity induced by a single vortex particle to the strength of the vortex particle. The spectra of the simulated velocities have been observed to be greater than their target, inspiring an investigation of the effect of varying K. The variance of the sampled horizontal velocity signal σ_u^2 is below target and above target for the vertical velocity signal σ_w^2. This corresponds to the deviations in the low frequency range of S_{uu} and S_{ww} respectively, see Fig. 2.5. Outside the low frequency range the spectra are generally above the target by a constant factor ≈ 1.25. However, the distribution is correct indicating that the conversion from grid velocities to particles strengths is valid in this region and that the offset is caused by the circulation correction K. By varying K the spectra can be offset uniformly by a factor as seen in Fig. 2.6. Thus, K can be adjusted to achieve better agreement between the spectral density of the sampled velocity and the target. As expected the variance scales proportionally with K^2, hence (σ_u, σ_w) equals (4.87,4.17), (6.13, 5.36), and (11.0, 8.23) for $K = 0.70\pi/2$, $0.85\pi/2$, and $1.00\pi/2$, respectively.

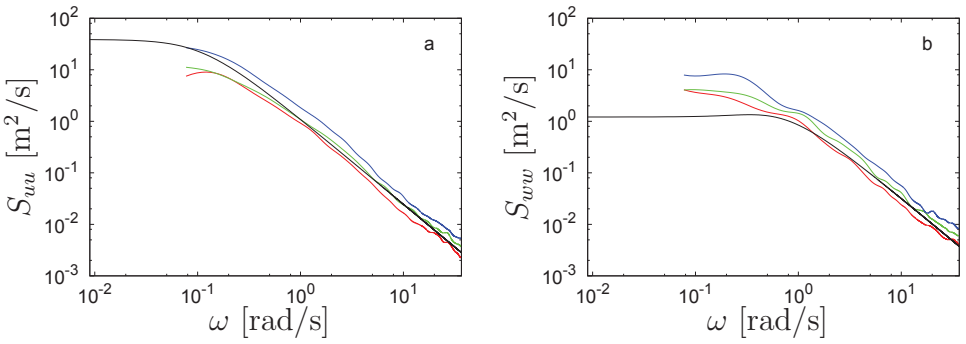


Figure 2.6: Spectral density of the (a) horizontal velocity S_{uu} and (b) vertical velocity S_{ww} for circulation integral correction (Eq. (2.2.20)) of $K = 0.70\pi/2$ (red),$K = 0.85\pi/2$ (green),$K = 1.00\pi/2$ (blue) and target (black). The spectra have been plotted in the range of the discrete frequencies used for turbulence generation.

2.4.4 Spatial dependency

The velocity field induced from a vortex particle is purely tangential relative to the position of particle cf. Eq. (2.1.11). Hence, the horizontal velocity component at a point is mainly induced by particles vertically aligned with the point and vice versa. Ideally a sample point should be surrounded with a larger number of vortex particles to ensure convergence of the turbulent energies of both the horizontal and vertical components. However, a finite distance to the edge of the particle band is sufficient as other effects, such as viscous diffusion, become more dominant than the contribution from far-field particles. This effect is studied by considering the variation of the spectral energy as a function of the position relative to the release grid. This is done by sampling at different positions on the centerline of the particle band, downstream of the release grid, $x = 1C, 2C, 4C, \ldots, 128C$. As the horizontal velocities mainly depend on particles vertically aligned with the sampling point, the energy σ_u^2 increases and converges quickly with respect to the downstream position x (not shown). Due to the strong dependency of the vertical velocity to the particles downstream and particularly upstream, σ_w^2 increases with x and converges approximately 16 times farther downstream than σ_u^2, at $x = 32C$ (not shown).

The influence of x on σ_w^2 is predominant and the increased energy is located mainly in the low frequency range of S_{ww} cf. Fig. 2.7(b). Immediately downstream of the release grid, and until $x \approx 8C$ (not shown), the energy of the high frequency ranges grows from approximately 70 % below

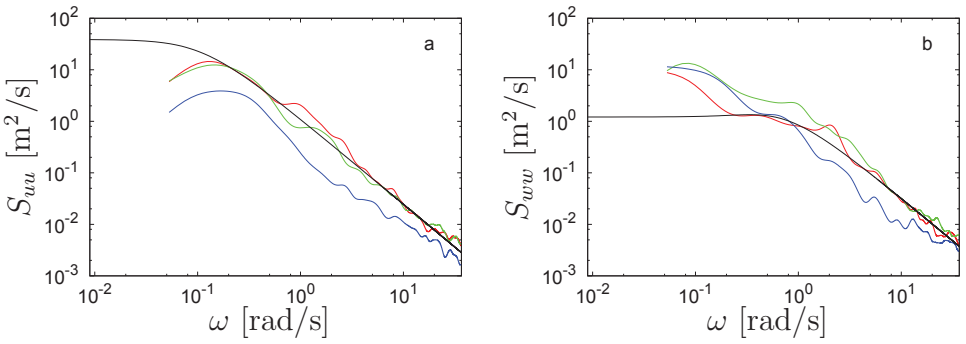


Figure 2.7: Spectral densities (a) S_{uu} and (b) S_{ww} of the velocity field sampled on the particle band centerline: $2C$ (red), $32C$ (green), $128C$ (blue) downstream of the release grid. The height of the grid is $12C$. The signal has been subsampled twice and the spectra smoothened by N-point Béziers. The spectra have been plotted in the range of the discrete frequencies used for turbulence generation.

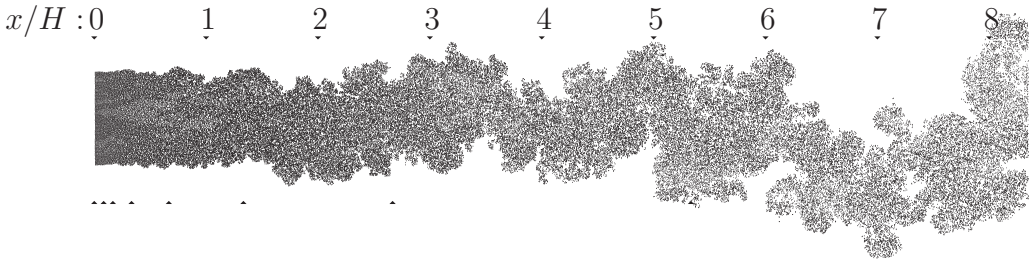


Figure 2.8: Flow visualization in the range from $x = 0C$ to $x = 100C$. The downstream distance is marked in units of particle band heights H (above). Below the particle band the x-coordinates of the sampling points of section 2.4.4 have been marked. The height of the release grid is $H = 12C$.

to 70 % above target. At this point the spectrum has converged for all but $\omega < 0.25$, in which frequency range the energy grows continuously as the flow is convected. For sampling points in the range $16C$–$45C$, S_{ww} remains unchanged and the sampling point can be considered sufficiently immersed in the turbulent flow, and sufficiently far from the rotation-free flow upstream of the release grid. After $45C$ the spectral energy gradually decreases. Initially in for the mid range frequencies but eventually extending to the entire spectrum. Fig. 2.8 indicates that this is due to mixing of rotation free flow towards the centerline of the particle band.

The standard deviation of the horizontal fluctuations σ_u^2 is below target which is primarily due to the energy of the low frequency range of S_{uu}. It is

conjectured that a higher particle band would cause a similar deviation of S_{uu} to the target as that of S_{ww}, i.e. stronger low frequency energy. Due to memory constraints the particle band height of the reference flow is limited to $12C$, i.e. to 15 % of the distance required for σ_w^2 to converge. S_{uu} converges at $x = 2C$, considerably faster than S_{ww}. As the flow is convected downstream the horizontal turbulent energy diminishes as shown in Fig. 2.7(a). From $x = 8C$ the energy of the mid range frequencies decreases slightly and further downstream, at $x = 64C$ the entire frequency range of S_{uu} decays almost uniformly.

The aerodynamic admittance has a stronger dependence on S_{ww} than on S_{uu} and any solid objects to be investigated in the present work is placed with the leading edge at $x = 20C$. Hereby the object is immersed in a region of the flow with constant spectral properties and a representative velocity signal can be sampled upstream of the object. When sampling $4C$ upstream of the leading edge neither the plate nor the Great Belt East bridge section has any significant influence on the spectral densities of the sampled velocities.

Sampling will be performed on the centerline of the particle band. Sampling off of the centerline of the $12C$ high particle band at $x = 16C$ shows only insignificant influence when e.g. sampling $4C$ off the centerline, i.e. $2C$ from the edge of the particle band (not shown). The deviation is only for the very lowest frequency range and mainly in the horizontal velocity spectrum S_{uu}. This suggests that the finite grid height $H = 12C$ leaves a broad vertical margin in which the flow is spectrally uniform. This does not mean that a particle band height of $2 \times 2C$ is sufficient as will be seen in section 2.4.5.

2.4.5 Dependency of grid height

The band of particles must be of a certain height for a structure to be exposed to a flow with the properties of atmospheric turbulence. The required height of the particle band is investigated by keeping the insert interval Δ_p fixed thereby not changing the particle density. The height of the release grid is varied from from $3C$ to $12C$ corresponding to 30 to 120 vortex particles per release.

The energy of the horizontal turbulent fluctuations increases due to the influence of the higher band of upstream particles. As Fig. 2.9(a) indicates increasing the grid height causes a consistent increase of turbulent energy in the low frequency range of the energy spectrum. This indicates that the height of the particle band limits large low frequency structures.

There is not a similar consistent relation between the magnitude of the vertical turbulent energy and the grid height, as shown in Fig. 2.9(b). The

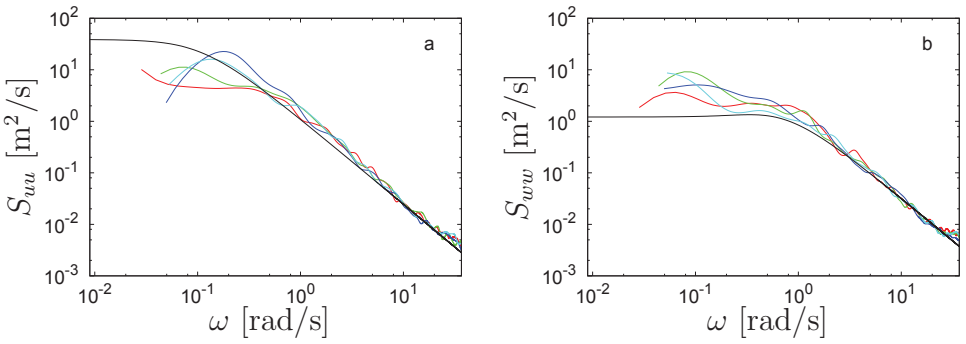


Figure 2.9: Spectral density for the (a) horizontal S_{uu} and (b) vertical S_{ww} velocities, for varying release grid height $3C$ and 30 points (red), $6C$ and 60 points (green), $9C$ and 90 points (blue), $12C$ and 120 points (cyan). The signal is sampled $10C$ downstream of the release grid and has been subsampled five times and the spectra averaged. The spectra have been plotted in the range of the discrete frequencies used for turbulence generation.

vertical turbulent energy σ_w^2 grows downstream of the release grid until it converges. For all grid heights σ_w^2 starts to decay at approximately 3.5 times the grid height for the present settings. This corresponds well to the mixing of rotation free flow towards the centerline of the particle band as shown in Fig. 2.8.

2.4.6 Dependency of inter-particle spacing

The band of particles must be sufficiently densely filled and the distance between particles at the release grid is determined by the width of the release grid cells. As described in section 2.2 the width of the cells is proportional to the free-stream velocity and the interval with which particles are inserted into the stream Δ_p. Thus Δ_p controls the density of particles and by decreasing the particle insert interval Δ_p, particles are being inserted at a higher rate. Since the cells of the release grid are quadratic, the particles will be more closely spaced in both the streamwise and vertical directions. The number of particles per release strongly influences the computational requirements, the height of the particle band. Thus to be able to set $\Delta_p = 1$ a particle band of height $3C$ is used with the entailing energy deficiencies described in section 2.4.5.

By varying the rate with which particles are released into the flow it has been observed that an increased energy in the high frequency range follows an increase of the particle insert interval Δ_p. When sampling velocities

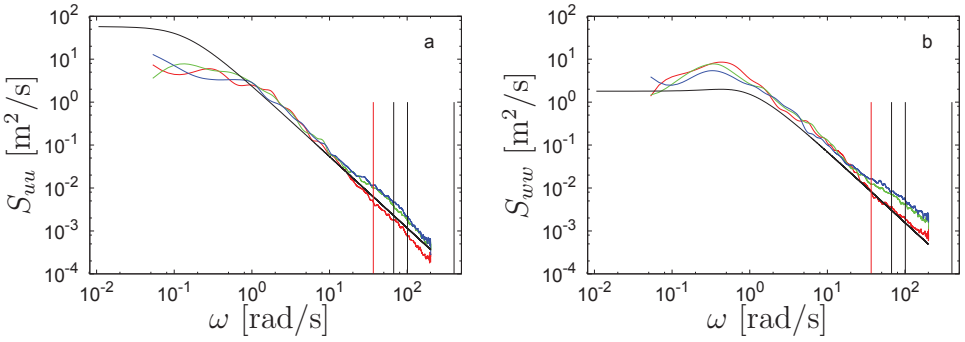


Figure 2.10: Spectral density of the (a) horizontal velocities S_{uu} and (b) vertical velocities S_{ww} sampled $10C$ downstream of grid. The insert interval is varied between 1 (red), 4 (green), and 6 (blue). The vertical red line marks ω_{u}, the black lines mark $\omega_{release}$ corresponding $\Delta_p = 6$, $\Delta_p = 4$, $\Delta_p = 1$ from left to right. Due to memory constraints a fixed release grid height of $\approx 3C$ has been used to allow insertion every time step ($\Delta_p = 1$). The spectra are averages of 5 subsamples and have been smoothened by an N-point Bézier.

directly at the release grid the frequency of the particle release

$$\omega_{release} = \frac{2\pi}{\Delta t \Delta_p}, \tag{2.4.2}$$

is visible as a well-defined spike. Further downstream the energy of the spike spreads to the surrounding frequency range obscuring the spectral position of $\omega_{release}$. At $x = 10C$ the positions of the energy spikes corresponding to $\Delta_p = 4$ and $\Delta_p = 6$ can no longer be distinguished, but the energy has spread to frequencies lower than the upper cutoff frequency ω_{u} as shown in Fig. 2.10. $\omega_{release}$ corresponding to $\Delta_p = 1$ is outside the visible range. Setting $\Delta_p = 1$ removes the artificial high frequency energy for $\omega < \omega_{\mathrm{u}}$ and though this gives better agreement with the respective targets, it restricts the height of the release grid due to memory constraints. Furthermore the added energy is far above the frequency range of interest when looking at cable bridges or pedestrian bridges. In spite of the deviation of the high frequency range of the spectra the aerodynamic admittance shows good agreement with target cf. section 2.5. In the present work large cable bridges are of interest and $\Delta_p = 4$ is chosen as this gives little deviation of the spectra in the high frequency range and ensures a sufficient height of the particle band.

Velocities and particles could be generated on a non-uniform grid. By creating particles of varying size (with an integer ratio) particles may be inserted more sparsely far off of the grid centerline, improving in particular

the vertical velocity spectrum. By evaluating the velocity processes at positions off the grid nodes (even outside the grid) larger cells may be integrated or higher order numerical integration may be used to evaluate the particle circulation. These options have not been pursued in the present work.

2.5 Aerodynamic admittance of a flat plate

2.5.1 Comparison with analytic solution

The flow past an infinitely thin plate is well studied in potential flow theory [102, 107], and by approximating the potential flow conditions the plate serves as a suitable benchmark. By assuming the vertical fluctuations are small compared to the mean speed of the flow the admittance has been approximated by Liepmann [66]

$$\chi_L = \frac{1}{1 + (\pi C/U)\omega}. \tag{2.5.1}$$

In the present study, the potential flow past an infinitely thin plate subjected to an oncoming turbulent flow is approximated by the viscous flow past a flat plate of finite length and thickness. The viscous diffusion is modelled using random walks and hence the turbulent velocity fluctuation should be above a certain level for the turbulent velocity fluctuations to dominate the fluctuations of the viscosity modelling at the solid surface. Due to the finite thickness of the plate, the viscous flow and the turbulent fluctuations causing instantaneous angles of attack of up to 12 degrees, separation occurs around the plate, see Fig. 2.11. In the present work a plate thickness $D = 1/200C$ has been used. Since the physical properties of the numerical simulation do not fully correspond to those of the potential flow solution, some deviation is anticipated. The measured slopes of the lift (C'_L) and pitching moment (C'_M) are $C'_L = 5.5$ and $C'_M = -1.18$, respectively. The experimental values [56] obtained for a plate with a chord-to-thickness ratio of $C/D = 16$ are $C'_L = 5.8$ and $C'_M = -1.43$, and thus a deviation less than 5 % and 17 %, respectively.

Fig. 2.12 shows the spectral densities for the horizontal (S_{uu}) and vertical (S_{ww}) velocities, the lift force S_{LL} and the pitching moment S_{MM}, as well as the corresponding aerodynamic admittance of the lift force χ_L and pitching moment χ_M. The velocity spectra agree with the results obtained in section 2.4. The lift force spectrum S_{LL} and pitching moment spectrum S_{MM} have been plotted with the predicted spectra from the frequency-independent relation, Eqs. (1.2.11) and (1.2.12) as reference. The reference spectra are

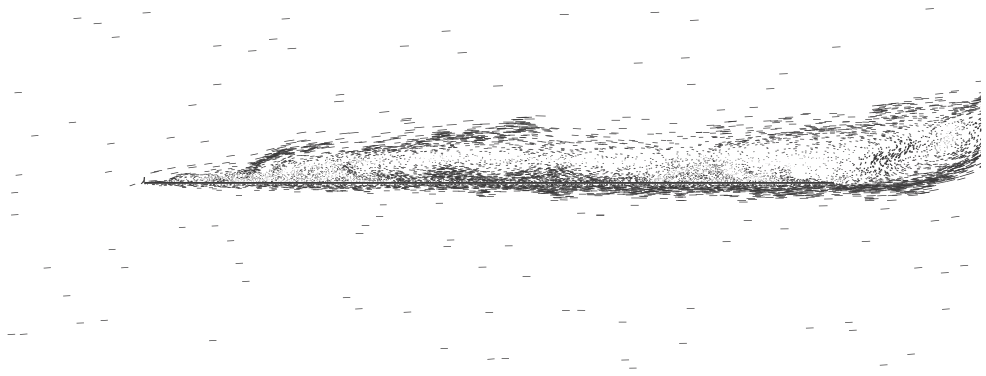


Figure 2.11: Flow visualization of turbulent flow past the flat plate at Re=10,000 and $I_w = 5\,\%$. The turbulent fluctuations result in flow separation.

based on the assumption of a frequency-independent admittance and from Figs. 2.12(c) and 2.12(d) it is seen that the spectra cross the reference spectra. Except from the area of intersection of the spectra, neither their magnitudes nor their slopes match. In spite of the aerodynamic admittance generally being frequency dependent the frequency-independent assumption is widely used as an initial approximation. In the low frequency range an increase of the spectral energy, similar to that of S_{ww}, can be seen in both S_{LL} and S_{MM}. This indicates that the forces on the plate are reactions to these added low frequency components of the flow. In spite that the vertical spectrum deviates from the prescribed turbulence spectrum, the aerodynamic admittance of both the lift force χ_L and the pitching moment χ_M are in reasonable agreement with Liepmann's approximate solution. The computed admittances χ_L and χ_M are generally 75 % higher than the approximation. In the low frequency range the computed admittances deviate below the profile of Eq. (2.5.1). This is conjectured to be due to large low frequency turbulent flow structures resulting in instantaneous angles of attack outside the valid range of Eq. (1.2.2). For frequencies above 10 rad/s the deviation of the computed admittances to the analytic solution increases further. It is recalled that comparison is performed with the potential flow solution entailing the above mentioned requirements.

In experiments the aerodynamic admittance has been found to depend on the spectral density of the turbulent fluctuations [57]. When generating the wind tunnel turbulence by spires the measured admittance is generally above Liepmann's approximation (Eq. (2.5.1)). However, at the lowest frequencies the measured admittance is considerably below Eq. (2.5.1). It is conjectured that the stronger admittance (and in turn the lift signal) is due to body

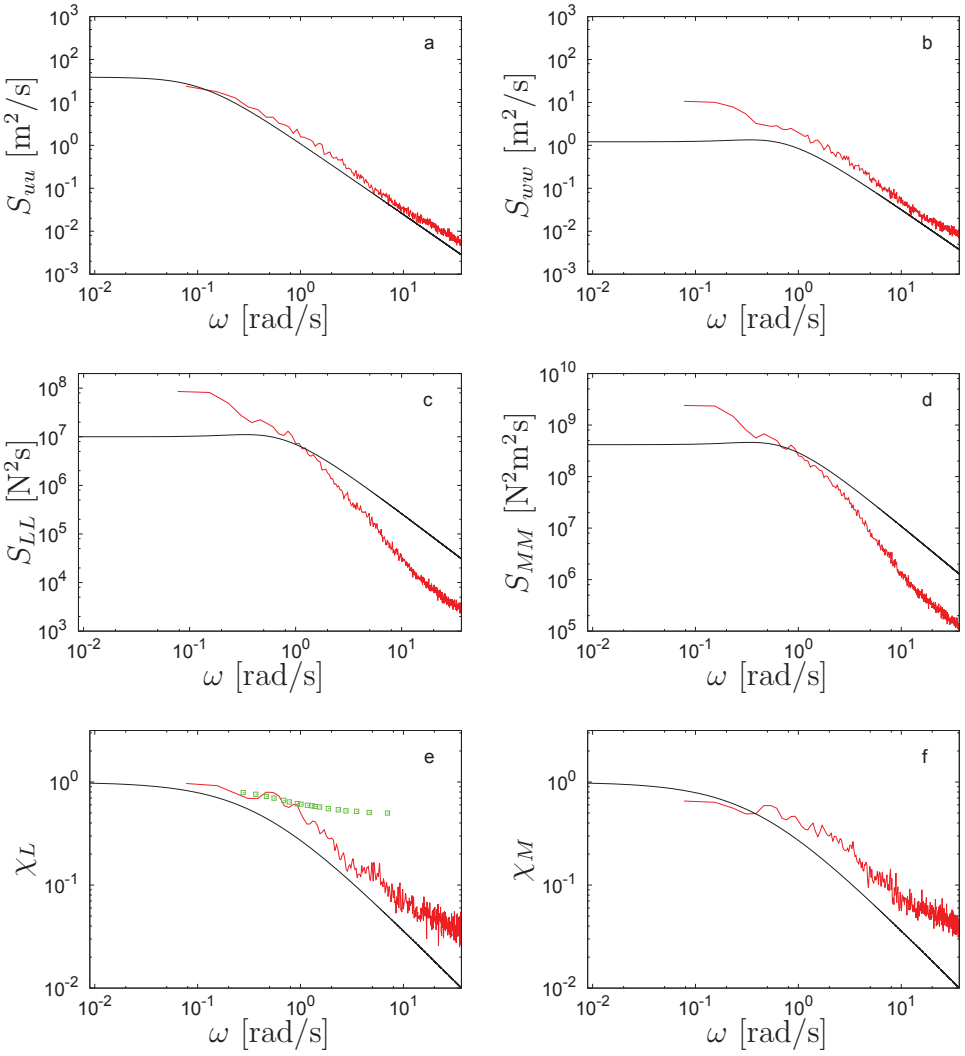


Figure 2.12: Spectral densities (red) (a) S_{uu} and (b) S_{ww}, and their target (black). (c) S_{LL} and (d) S_{MM} are spectral densities of the lift force and pitching moment (red) measured on the flat plate and the predictions (black) by Eqs. (1.2.11) and (1.2.12). Aerodynamic admittance of the lift force χ_L(red) and experimental results [56] for a $C/D = 16$ plate (□), (e) and the pitching moment χ_M (f) for the flat plate (red), compared to Liepmann's approximation [66] (black). The graphs are based on 380,000 samples subsampled 100 times. The spectra have been plotted in the range of the discrete frequencies used for turbulence generation.

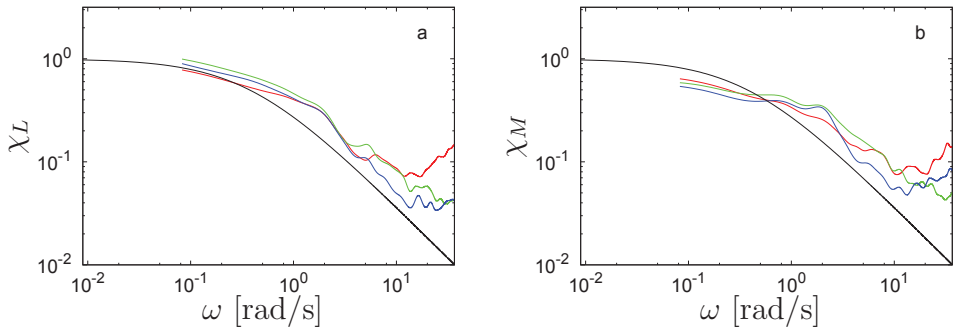


Figure 2.13: Aerodynamic admittance of the lift force χ_L (a) and the pitching moment χ_M (b) for the flat plate compared to Liepmann's approximation (black). Results from simulation at Re = 1000 (red), Re = 10000 (green), Re = 100000 (blue). The signals have been subsampled 10 times. The spectra have been plotted in the range of the discrete frequencies used for turbulence generation.

induced turbulence [57]. Both tendencies can be seen at Fig. 2.12(e) and 2.12(f). In the present work the flow around the plate separates due to the turbulent gusts as shown in Fig. 2.11. Though the ratio of chord to thickness $C/D = 12.7$ for the bridge section experiments [57] is lower than $C/D = 200$ for the flat plate the agreement to of Liepmann's approximation of the thin plate is better for the experimental results. However, the rectangular leading edge of the flat plate used in the present study is blunt, which may increase the aerodynamic admittance as demonstrated experimentally for a $C/D = 16$ plate [56] cf. Fig. 2.12(e).

2.5.2 Influence of Reynolds number and turbulence intensity

The deviation in the high frequency range of the computed admittances to the analytic solution is suspected to be caused by the viscosity modelling. That is, the standard deviation of the diffusion step length, Eq. (2.1.15), influences the magnitude of the high frequency deviation. The reference case is simulated at different Reynolds numbers, by varying viscosity and thereby the average diffusion step lengths. As shown in Fig. 2.13 the high frequency range of both the lift force admittance χ_L and the pitching moment admittance χ_M decrease with increasing Reynolds number.

As seen in section 2.4.3 increasing the strength of the particles inserted to generate turbulence increases turbulence intensity. Contrary to expectation the high frequency range deviation is not increased as shown in Fig. 2.14.

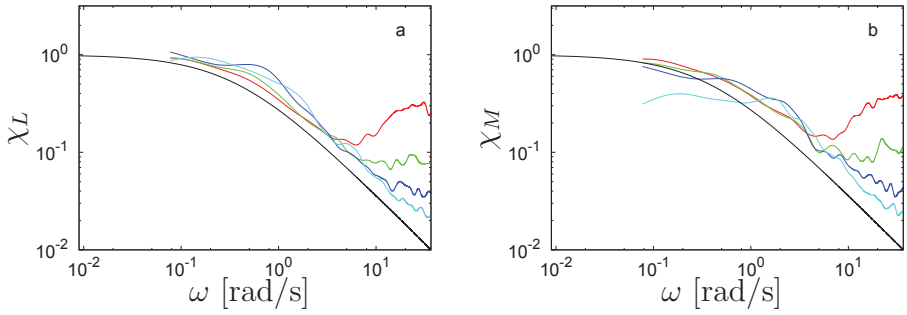


Figure 2.14: Aerodynamic admittance of the lift force χ_L (a) and the pitching moment χ_M (b) for the flat plate compared to Liepmann's approximation (black). By varying the specified turbulence intensity from $I_w = 1.25\,\%$ (red), $I_w = 2.5\,\%$ (green), $I_w = 5.0\,\%$ (blue), $I_w = 10\,\%$ (cyan) it is seen that the influence from viscous diffusion at the surface becomes less dominant. The signals have been subsampled 10 times. The spectra have been plotted in the range of the discrete frequencies used for turbulence generation.

The deviation to the analytic solution is decreased considerably and for specified vertical turbulence intensities $I_w \geq 10\,\%$ the admittance assumes the shape of Eq. (2.5.1) from the mid-range frequencies up to ω_{u}. This indicates that the influence of the random walk to the force signal is solely related to the particles generated and emitted from the body surface to enforce the no penetration condition. For bluff bodies, vortex shedding appears in the admittance as a peak at the shedding frequency (see section 2.6.1).

2.6 Aerodynamic admittance of bridge sections and comparison to experimental results

Actual bridge sections may not be double or even single symmetric thus $C_L(\alpha)$ and $C_M(\alpha)$ may not behave linearly to the same extent as is the case for the flat plate, Fig. 2.15a. This causes C_L and C_M to depend differently on positive and negative α as Fig. 2.15 shows. Furthermore the linear region is narrower due to the increased separation of the blunter bridge sections. The negative $C_M(\alpha)$ slope has a stabilising effect on the pitching moment and thereby flutter. The linearisation of the angle of attack, Eq. (1.2.2), and the non-linear behaviour of the coefficients $C_L(\alpha)$ and $C_M(\alpha)$ constrains the conditions for which valid aerodynamic admittances can be computed.

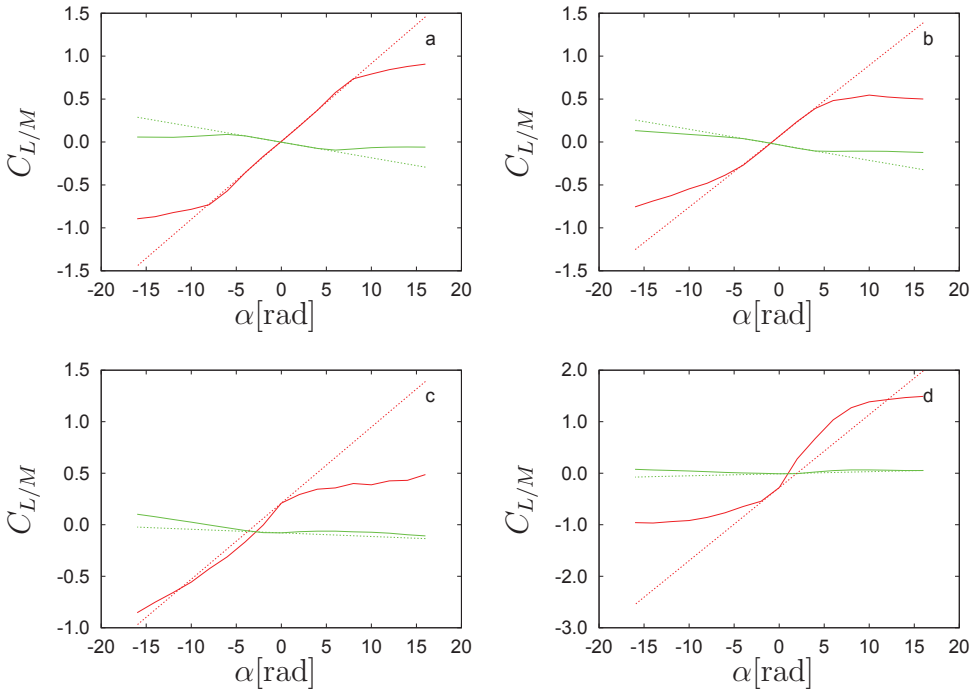


Figure 2.15: The lift (red) and pitching moment (green) coefficients of the bridge sections (solid) and the approximated linear fit (dashed) at various angles of attack. (a) Thin plate, (b) Great Belt East, (c) Øresund, (d) Busan-Geoje.

	C_L	C_M	C'_L	C'_M	ω_s
Great Belt East	0.07	-0.03	4.73	-1.04	10.48
Øresund	0.21	-0.08	4.23	-0.20	13.19
Busan-Geoje	-0.28	-0.01	8.12	0.13	8.14

Table 2.1: The static aerodynamic lift C_L and pitching moment C_M coefficients and the corresponding derivatives C'_L and C'_M for selected bridges. The shedding frequency ω_s is in rad/s.

Therefore some error must be expected. Tab. 2.1 shows the linearisation coefficients of the lift and pitching moment.

2.6.1 The Great Belt East bridge

Bridge sections are typically bluff bodies with increased vortex shedding compared to the flat plate. The thickness of the Great Belt East bridge section is $D = 0.14C$. Fig. 2.16 shows the aerodynamic admittance of the lift force χ_L and pitching moment χ_M as well as the corresponding spectral densities.

The velocity spectra are equal to those of the flat plate, as shown in Fig. 2.12. S_{LL} and S_{MM} for the bridge section is of the same order of magnitude as the frequency-independent assumption Eq. (1.2.11) and (1.2.12) in the frequency interval from 0.8 rad/s to 10 rad/s. This is due to vortex shedding. Therefore using the frequency-independent approximation as an initial estimate of the force spectrum may give reasonable results. Outside the interval the agreement is similar to that of the of the flat plate. Both χ_L and χ_M are stronger than Liepmann's approximation (2.5.1) and similar to the admittances of the flat plate. However, the vortex shedding is stronger for the bridge section which manifests itself as a peak at $\omega_s = 10.48$ rad in both χ_L and χ_M thereby deviating from the analytic solution that does not take into account vortex shedding. The snapshot of the flow simulation on Fig. 2.17 shows little separation along the bridge deck but vortex shedding at the trailing edge. As seen experimentally [57] the admittance below Eq. (2.5.1) at lower frequencies.

The turbulent variations are not periodic and have been found to reduce the periodic process of vortex shedding [105, 93]. Furthermore, by increasing the turbulence intensity the forces from the turbulent vertical fluctuations become dominant and the admittances tend to the analytic solution [66], see Fig. 2.18. Of course in the present simulations the bridge is fixed, thus there is no coupling back of the bridge motion to the flow which may in turn affect the aerodynamic admittance. The varying pressure distribution around the

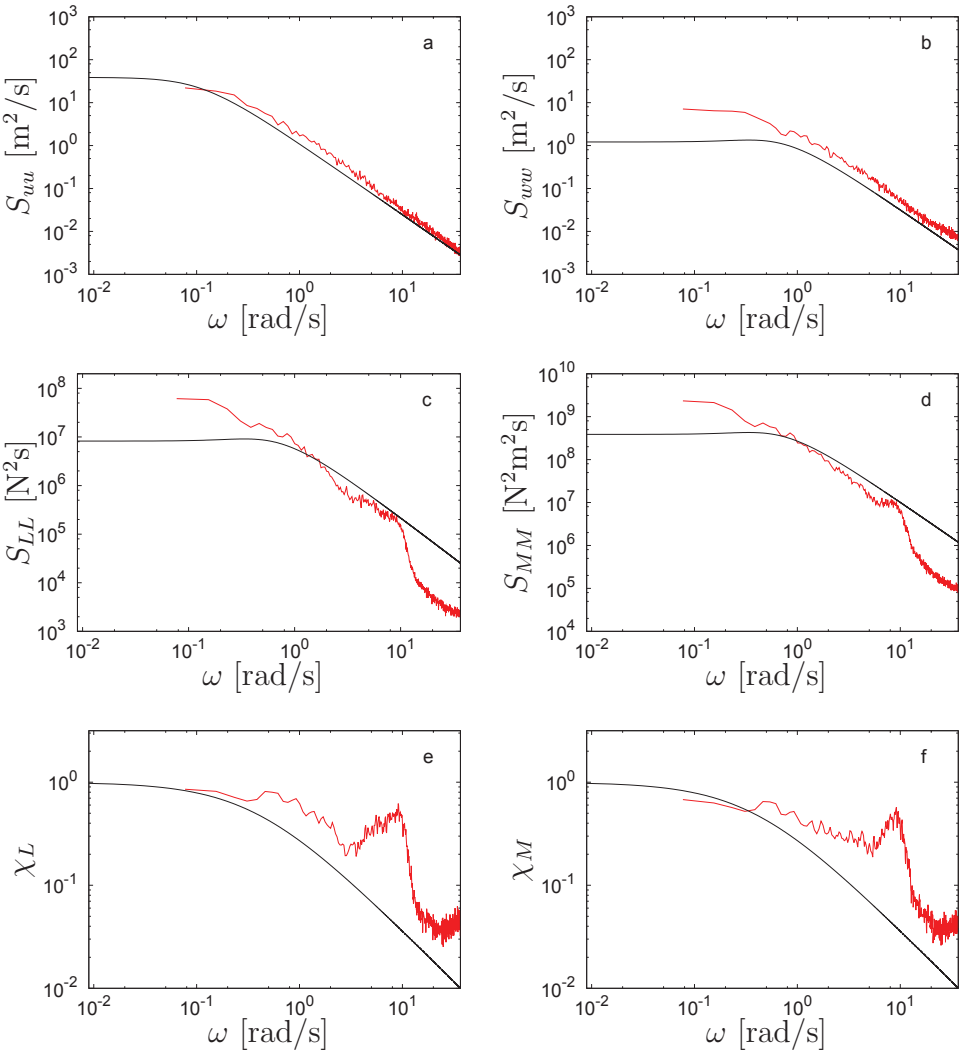


Figure 2.16: Spectral density (red) (a) S_{uu} and (b) S_{ww}, and their target (black). (c) S_{LL} and (d) S_{MM} are the spectral densities of the lift force and pitching moment (red) measured on the Great Belt bridge section and the predictions (black) by the frequency-independent assumption Eqs. (1.2.11) and (1.2.12). Aerodynamic admittance of the lift force (red) χ_L (e) and the pitching moment (red) χ_M (f) for the Great Belt bridge section, compared to Liepmann's approximation (black). The graphs are based on 380,000 samples subsampled 100 times. The spectra have been plotted in the range of the discrete frequencies used for turbulence generation.

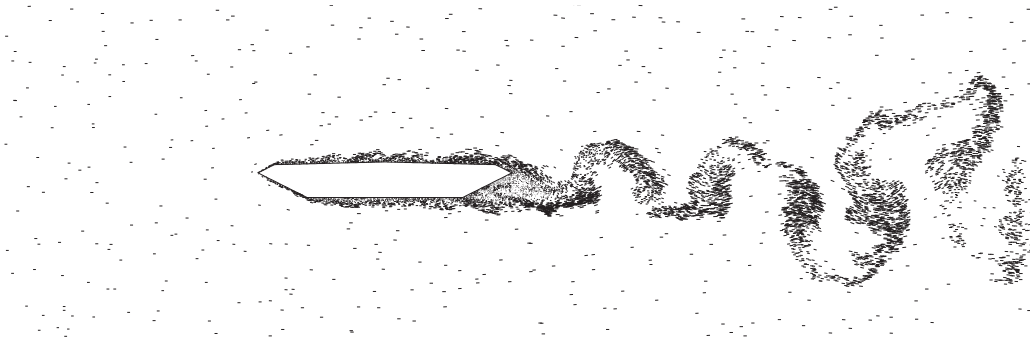


Figure 2.17: The simulated turbulent flow field past the Great Best East bridge. The illustration shows the instantaneous position and velocity of the individual vortex particles.

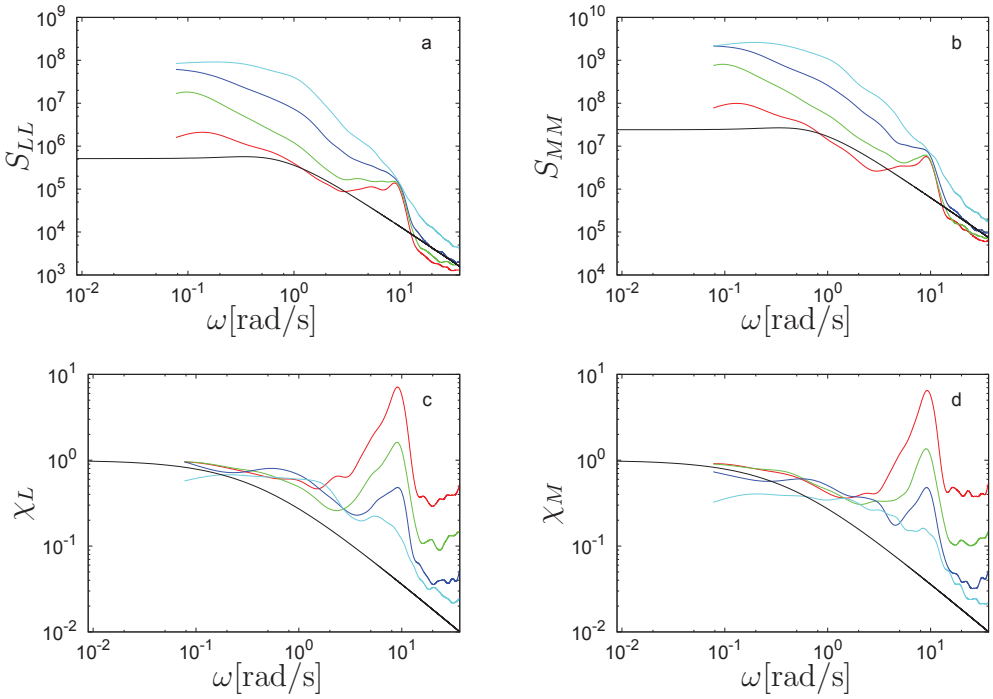


Figure 2.18: The spectral densities of (a) lift force $S_L L$ and (b) pitching moment $S_M M$ on the Great Belt East bridge. (c) and (d) the corresponding aerodynamic admittances. The specified turbulence intensity is varied from $I_w = 1.25\,\%$ (red), $I_w = 2.5\,\%$ (green), $I_w = 5.0\,\%$ (blue), $I_w = 10\,\%$ (cyan). The reference [66] (black). The spectra are averages from 10 subsamples and have been smoothened by N-point Béziers. The spectra have been plotted in the range of the discrete frequencies used for turbulence generation.

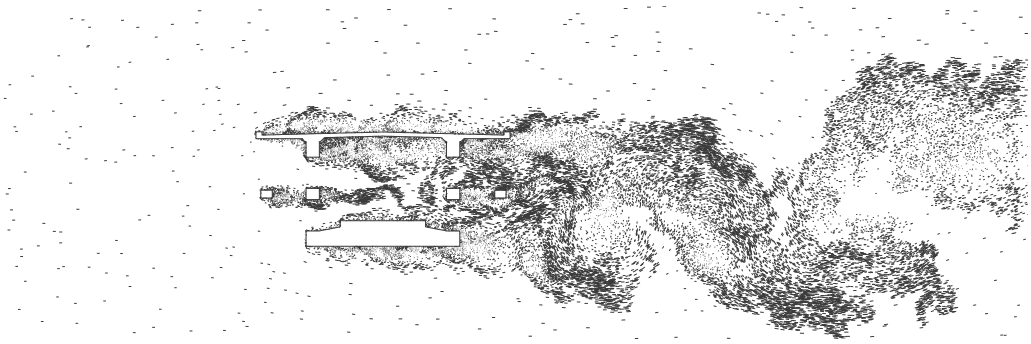


Figure 2.19: The simulated turbulent flow field past the Øresund bridge. The illustration shows the instantaneous position and velocity of the individual vortex particles.

bridge section is, like the turbulent velocity, not periodic and the response due to this force may in turn dampen the bridge motion induced by periodic forces.

2.6.2 The Øresund bridge

The Øresund bridge is a double deck truss bridge with a bridge section geometry. As seen on the flow visualisation, Fig. 2.19, the surface to volume area is high. Thereby the turbulent forces work on a larger total area and on substructures of different magnitude thus increasing the admittance at multiple frequency ranges. Also the flow visualisation imply that the vortex shedding of all substructures may contribute significantly. The computed aerodynamic admittances are significantly higher than Liepmann's approximation and frequency-independent approximation Eqs. (1.2.11) and (1.2.11), as seen on Fig. 2.20. However, the computed aerodynamic admittance function shows excellent agreement to that calculated from wind tunnel tests [100]. From the simulated aerodynamic admittance the vortex shedding frequency is identified at $\omega_s = 13.19$. This frequency cannot be identified from the wind tunnel data due to the measurement restrictions on the frequency range.

2.6.3 The Busan-Geoje bridge

The Busan-Geoje bridge is a plate girder bridge as seen on the flow visualisation Fig. 2.21. The flow recirculates between the vertical sides which is know to stimulate flutter [65]. The original Tacoma Narrows bridge [35], the Long Creeks bridge in Canada [48] and the Kessock bridge in Scotland

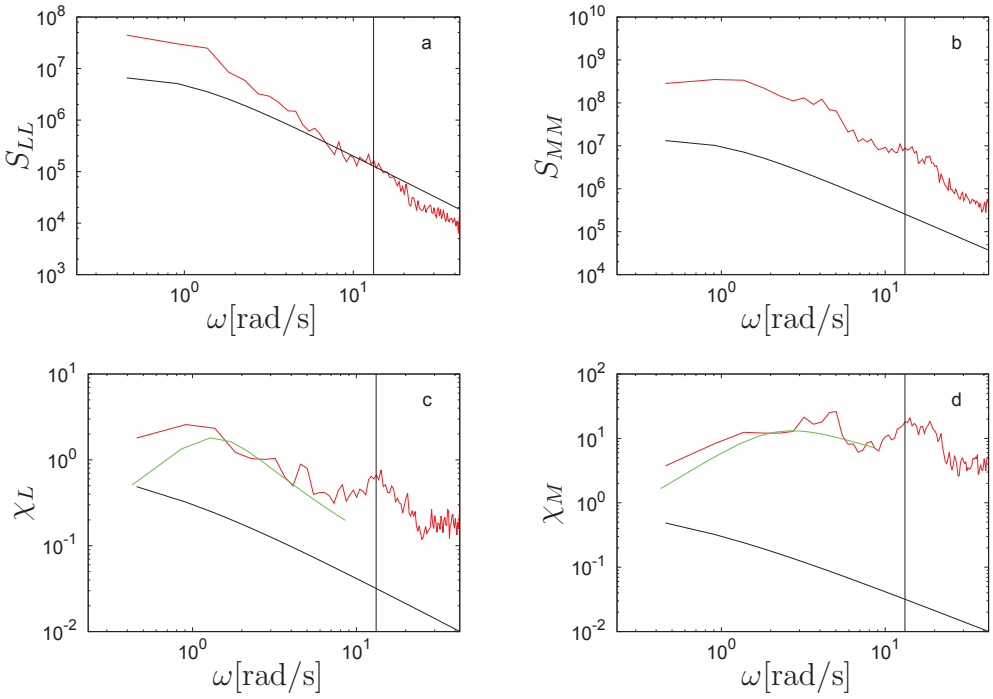


Figure 2.20: Spectral densities and admittances based on flow simulation around the Øresund bridge. (a) Lift force spectrum (red) and (b) pitching moment spectrum (red) and the prediction spectra according to the frequency-independent assumption Eqs. (1.2.11) and (1.2.12) (black). The admittance of (c) the lift force and (d) the pitching moment are significantly higher than Liepmann's approximation (black) but match experiments (green) well.

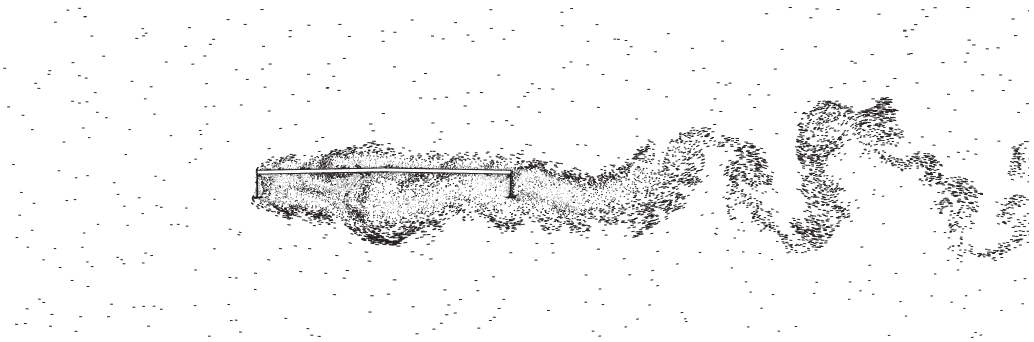


Figure 2.21: The simulated turbulent flow field past the Busan-Geoje bridge. The illustration shows the instantaneous position and velocity of the individual vortex particles.

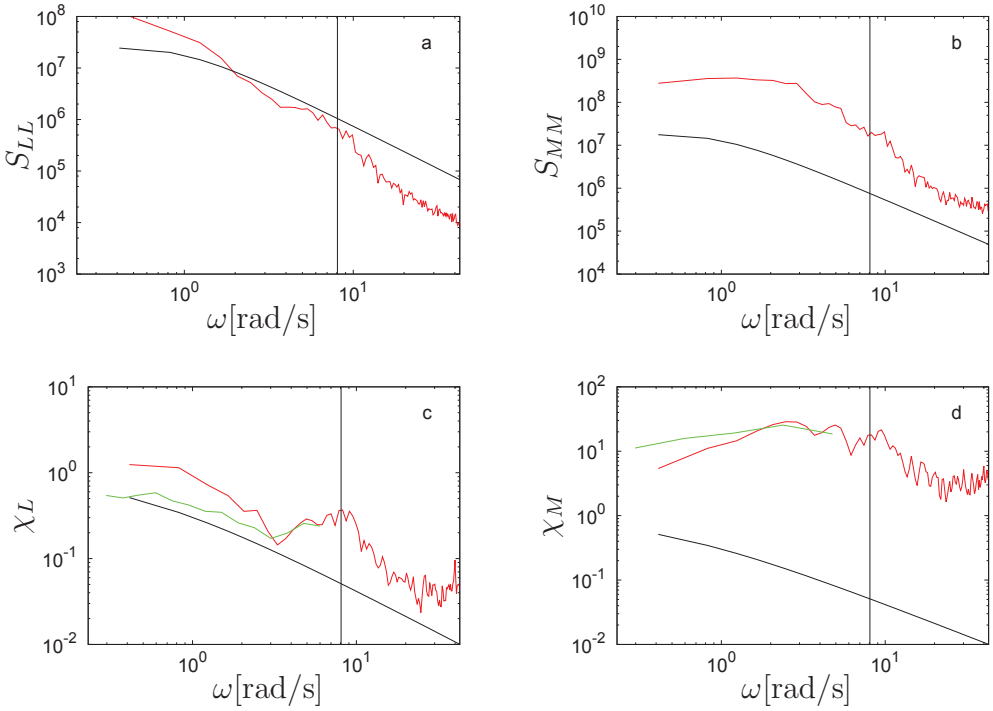


Figure 2.22: Spectral densities and admittances based on flow simulation around the Busan-Geoje bridge. (a) Lift force spectrum (red) and (b) pitching moment spectrum (red) and the prediction spectra according to the frequency-independent assumption Eqs. (1.2.11) and (1.2.12) (black). The admittance of (c) the lift force and (d) the pitching moment are higher than Liepmann's approximation (black) but match experiments (green) well.

[48], all with geometries resembling that of the Busan-Geoje bridge, suffered from extensive pitching movements. Fig. 2.22 shows that the aerodynamic admittances are higher than Liepmann's approximation. In particular the aerodynamic admittance of the pitching moment which is also higher than the frequency-independent approximation Eqs. (1.2.11) and (1.2.11). Both aerodynamic admittances agree very well to the experimental data.

2.6.4 Summary

Based on the two-dimensional discrete vortex method, turbulent oncoming flow is introduced through the insertion of upstream vortex particles modelling the anisotropic turbulent velocity spectra. The admittances of the lift and pitching moment are obtained from the measured spectra of the turbulent flow field and the corresponding spectra of the aerodynamic loads. The

method has been validated through detailed simulations of the turbulent flow past a flat plate and past the Great Belt East bridge, the Øresund bridge and the Busan-Geoje bridge. The results were found in good agreement with the semi-analytical model of Liepmann and wind tunnel experiments. The method is expected to be a useful engineering tool in bridge aerodynamics.

Chapter 3

Multiresolution particle-mesh method

The Vortex-In-Cell (VIC) method [7, 17] is a highly efficient hybrid particle-mesh algorithm for the simulation of incompressible viscous flow. Using fast Fourier transforms (FFT) the algorithm scales $\mathcal{O}(N \log N)$ with the number of computational elements but constrains the velocity computations to uniformly resolved meshes. The present work introduces a novel multiresolution to the VIC method solution, thereby increasing its efficiency.

3.1 The vortex-in-cell algorithm

The Vortex-In-Cell algorithm [7, 17] evolves the incompressible flow described by the vorticity-transport equation

$$\frac{D\omega}{Dt} = \nu \boldsymbol{\nabla}^2 \omega \tag{3.1.1}$$

with constant kinematic viscosity ν. Eq. (3.1.1) describes the rate of change of vorticity

$$\boldsymbol{\omega} = \boldsymbol{\nabla} \times \boldsymbol{u} \tag{3.1.2}$$

of a particle moving with the velocity $\boldsymbol{u}$. The vorticity in two-dimensional analysis is a scalar ω equal to the out of plane component ($\boldsymbol{e}_z$) of the curl of the velocity field, $\boldsymbol{\omega} = \omega \boldsymbol{e}_z$. The velocity $\boldsymbol{u}$ is defined as

$$\boldsymbol{u} = \boldsymbol{\nabla} \times \boldsymbol{\Psi} + \boldsymbol{U}_\infty \tag{3.1.3}$$

where $\boldsymbol{\Psi}$ is a stream function and $\boldsymbol{U}_\infty$ the irrotational free stream velocity. Combining Eq. (3.1.3) with the definition of vorticity (Eq. (3.1.2)) yields the

Poisson equation

$$\boldsymbol{\nabla}^2\Psi = -\omega \tag{3.1.4}$$

linking vorticity to the stream function and in turn to the velocity, cf. Eq. (3.1.3). The stream function is calculated on the mesh from the Poisson equation, Eq. (3.1.4), using a Green's function solution [46]

$$\Psi(\boldsymbol{x}) = \int_V G(\boldsymbol{y}-\boldsymbol{x})\,\omega(\boldsymbol{x})\,\mathrm{d}\boldsymbol{y} \tag{3.1.5}$$

$$\Psi = G * \omega \tag{3.1.6}$$

where

$$G(\boldsymbol{y}-\boldsymbol{x}) = -\frac{1}{2\pi}\log|\boldsymbol{y}-\boldsymbol{x}|$$

is the free-space Green's function to the Poisson equation in two dimensions [4]. The singularity is replaced by $G(\mathbf{0}) = 1$ [46]. The convolution is performed in Fourier space according to the convolution theorem. The Fourier transformation is carried out numerically using the fast Fourier transform which entails a periodicity of G, ω and in turn Ψ. The periodic representation entails circular convolution. That is, as G is shifted out of the domain during the convolution it re-enters at the opposite side as shown on Fig. 3.1a. To circumvent the periodicity the domain is doubled in each physical direction by padding the vorticity field ω by zeros [45, 46, 69]. The Green's function is extended to the new domain size thereby including the required additional far-field information. As the Green's function re-enters the domain its contribution is cancelled as it is applied to the zero-region, see Fig. 3.1b.

The vorticity field ω is approximated by N discrete vorticity-carrying particles. The particle vorticity is interpolated to the mesh points $\boldsymbol{x}_m$ of an underlying equally spaced Cartesian mesh, through

$$\omega_m = \sum_p^N \omega_p W\left(\frac{\boldsymbol{x}_p - \boldsymbol{x}_m}{\Delta x}\right) \tag{3.1.7}$$

using the third order M_4' [67] interpolation kernel

$$W(x) = \begin{cases} 0 & \text{for} \quad |x| > 2 \\ \frac{1}{2}(2-|x|)^2(1-|x|) & \text{for } 2 \geq |x| > 1 \\ 1 - \frac{5}{2}x^2 + \frac{3}{2}|x|^3 & \text{for } 1 \geq |x| \end{cases} \tag{3.1.8}$$

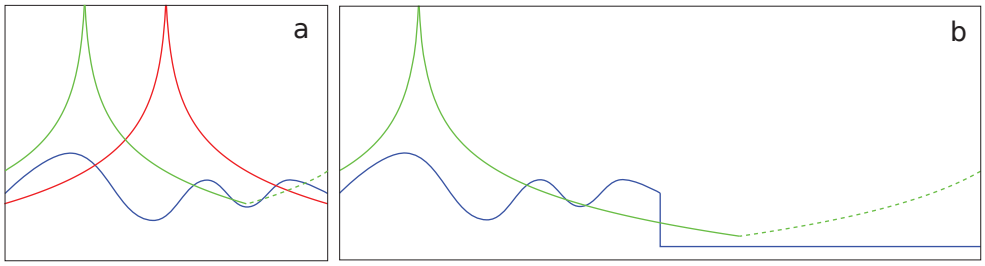


Figure 3.1: (a) Sketch of the convolution between the Green's function (red) and the vorticity (blue). Due to the periodicity of the FFT the convolution is circular. Thus the Green's function re-enters (green) when shifted during the convolution. (b) By doubling the domain and padding the vorticity with zeros the re-entering Green's function gives no contribution, cancelling the periodicity.

in each coordinate direction, conserving 0, 1^{st} and 2^{nd} order moments. Δx is the mesh spacing.

From the velocity $\boldsymbol{u}$ and vorticity ω on the mesh the right hand side of Eq. (3.1.1) is computed and interpolated back to the particles along with the velocity. The particle position $\boldsymbol{x}_p$ and vorticity ω_p is updated using Runge-Kutta second order time integration.

For the simulation of advection problems, particle-mesh algorithms are not constrained with the usual CFL condition. Instead, the time step size (Δt) is limited by the amount of shear, through a condition of the type

$$\Delta t < \frac{C_1}{|\omega|_\infty}$$

which does not directly depend on the spatial resolution. C_1 is a constant depending on the integration scheme. In the cases dealt with in the present work, the limiting factor is the explicit discretisation of the diffusion and the time step is chosen to satisfy

$$\Delta t < \frac{C_2 \Delta x^2}{\nu}.$$

$C_2 = 1/4$ is a constant depending on the integration scheme.

As the particles are advected they may be subject to numerical errors imposing compression/dilation in certain regions whereas other regions become sparsely populated. To ensure a sufficient resolution in the entire domain and to avoid strong deformation of the particle volumes new particles are remeshed from the mesh. Remeshing at every n_r time step improves spatial convergence [52]. Presently $n_r = 1$.

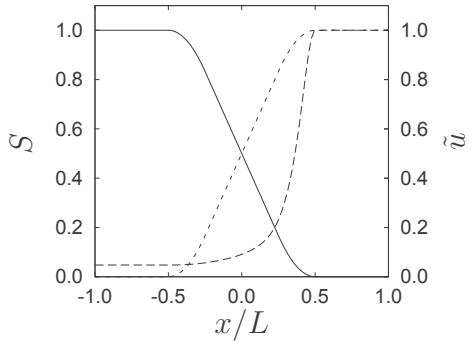


Figure 3.2: The polynomial step function S (———) as a function of the signed distance to the solid surface. Implicit penalization of a uniform velocity field $U = 1$ (Eq. (3.2.2)) results in velocity fields $\tilde{u}$ (- - - - -) with a residual of $1/\lambda\Delta t$ of U in the solid. (- - - - - -) is the velocity by interpolation penalization Eq. (3.2.4). For the sake of clarity $\lambda\Delta t = 20$.

3.2 Brinkman penalization

The no-slip condition at solid surfaces may be imposed by using Brinkman penalization [1, 50, 20]. By adding the Brinkman penalization term to the Navier-Stokes equation, Eq. (3.1.1) becomes

$$\frac{D\omega}{Dt} = \nu\boldsymbol{\nabla}^2\omega + \boldsymbol{\nabla} \times [\lambda\chi(\boldsymbol{u}_s - \boldsymbol{u})] \tag{3.2.1}$$

where λ is the penalization parameter ($[1/s]$), equivalent of an inverse porosity and $\boldsymbol{u}_s$ is the solid velocity. χ is the solid mask that is 1 inside the solid and 0 in the fluid. When evaluating the penalization Eq. (3.1.1) explicitly it may be necessary to deactivate the right hand side of the vorticity equation in the solid domain. In the present work χ is defined solely on the meshes. To improve the numerical accuracy of the curl, χ is varied smoothly from 0 to 1 over the mollification interval of width L normal to the surface, corresponding to a fixed number of mesh cells. χ is constructed via a mollified step function S, a function of the signed distance to the surface of the solid. In the present work S is determined such that it is continuous and differentiable and consists of second order polynomials in the first and last $L/4$ of the mollification interval and first order polynomial in the intermediate region, see Fig. 3.2. It is desirable to choose $\lambda T \gg 1$, where T is a flow characteristic time, but, in the case of explicit Euler time discretisation the penalization parameter must satisfy $\lambda\Delta t < \mathcal{O}(1)$ to ensure stability. Instead of evaluating the penalization as a source term, cf. Eq. (3.2.1), it can be evaluated semi-implicitly [20] by using a split-step algorithm. That is, at time

step n: update the vorticity field by evaluating the penalization term and subsequently evaluate the right hand side of Eq. (3.1.1). The penalization term can be evaluated implicitly by the expression

$$\tilde{\boldsymbol{u}}^{n+1} = \frac{\boldsymbol{u}^n + \lambda \Delta t \chi \boldsymbol{u}_s^n}{1 + \lambda \Delta t \chi} \tag{3.2.2}$$

$$\tilde{\omega}^{n+1} = \boldsymbol{\nabla} \times \tilde{\boldsymbol{u}}^{n+1} \tag{3.2.3}$$

where the tilde denotes that the result is not the final solution of the time step. This facilitates the use of high λ values but with an increasingly discontinuous velocity field as a consequence when using a smooth step function S. Also, a residual of $\boldsymbol{u}^n$ on the order of $1/\lambda\Delta t$ is left in the solid interior, cf. Fig. 3.2. Instead the penalization term may be evaluated explicitly in a similar split-step algorithm with $\lambda = 1/\Delta t$. The resulting expression may be viewed as simple interpolation between the velocity fields $\boldsymbol{u}^n$ and $\boldsymbol{u}_s^n$

$$\tilde{\boldsymbol{u}}^{n+1} = (1 - \chi)\boldsymbol{u}^n + \chi \boldsymbol{u}_s^n, \tag{3.2.4}$$

thereby eliminating the penalization parameter λ, removing the residual velocity and controlling the weight of the solid and fluid velocity explicitly. The *interpolating penalization* Eq. (3.2.4) also corresponds to the semi-implicit penalization (Eq. (3.2.2)) using a special mask

$$\tilde{\chi} = \min\left(\frac{\chi}{\lambda \Delta t (1 - \chi)}, 1\right),$$

that is capped at 1 to avoid singularities, where χ is the corresponding mask of Eq. (3.2.4).

Replacing the vorticity field by the finite difference evaluation of Eq. (3.2.3) introduces significant numerical diffusion of the vorticity field. Therefore the penalization step Eqs. (3.2.4) and (3.2.3) is formulated as a correction

$$\begin{aligned} \Delta \boldsymbol{u} = \tilde{\boldsymbol{u}}^{n+1} - \boldsymbol{u}^n &= (1 - \chi)\boldsymbol{u}^n + \chi \boldsymbol{u}_s^n - \boldsymbol{u}^n \\ &= \chi(\boldsymbol{u}_s^n - \boldsymbol{u}^n) \\ \Delta \omega &= \boldsymbol{\nabla} \times \Delta \boldsymbol{u} \qquad (3.2.5) \\ \tilde{\omega}^{n+1} &= \omega^n + \Delta \omega \qquad (3.2.6) \end{aligned}$$

such that ω^n is not replaced but updated by the correction Eq. (3.2.5) evaluated by finite differences.

The vorticity correction, Eq. (3.2.6), is local and therefore the velocity field computed from the corrected vorticity can in general not be expected to provide a solution that exactly satisfies no-slip on the entire boundary. This

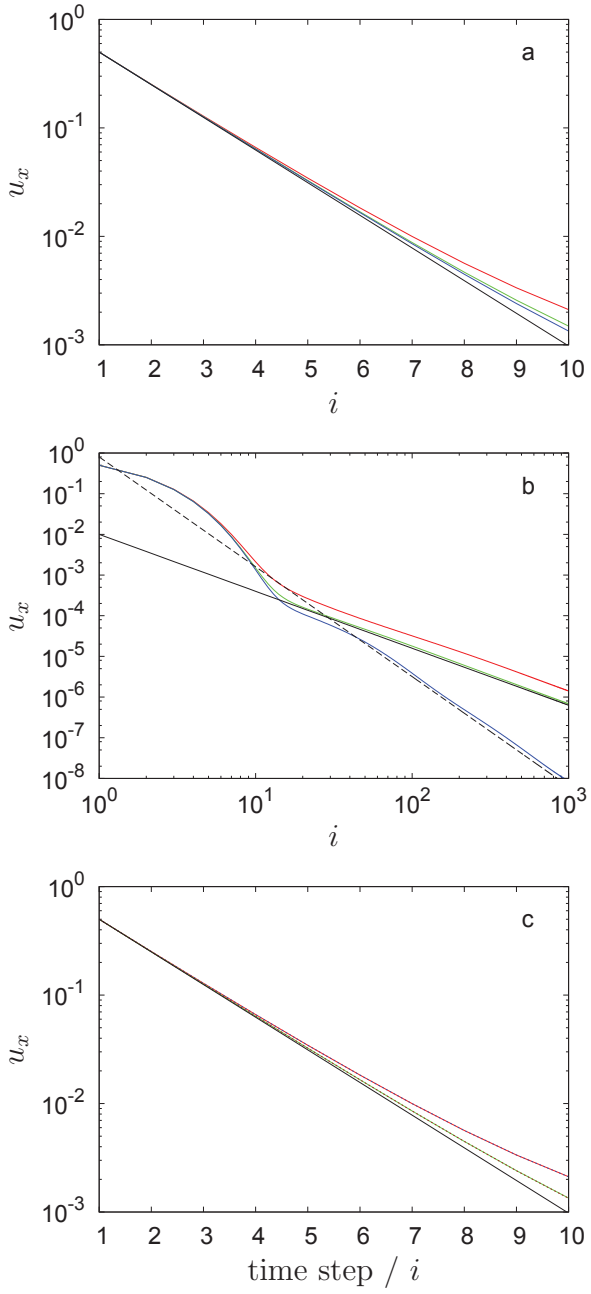


Figure 3.3: The remaining streamwise velocity component u_x/U in a static cylinder in an onset flow as a function of the number of penalization iterations i. The penalization layer has been varied between $3\sqrt{2}\Delta x$ (red), $\sqrt{2}\Delta x$ (green) and 0 (blue). The velocity decreases proportionally to the trend lines (a) 0.5^i (black) and (b) $i^{-1.4}$ (solid black) or $i^{-2.7}$ (dashed black). (c) Results from one penalization per time step (red/green, solid) coinciding with multiple penalization iterations (blue/brown, dashed) for penalization layer thickness ($3\sqrt{2}\Delta x$ / 0).

can be improved by repeating the penalization during a time step thereby increasing the correspondence between velocity and vorticity. Fig. 3.3a shows the streamwise velocity component in the centre of a static cylinder as a function of the number i of penalization repetitions. An onset flow with free-stream velocity U is applied and as Fig. 3.3a shows the velocity decreases nearly exponentially. The deviation to exponential decrease is slightly dependent on the width of the penalization layer. As Fig. 3.3b shows, this dependency becomes more predominant after 20 iterations where the convergence towards $\boldsymbol{u}_s$ becomes polynomial proportional. If no penalization layer is used the interior velocity converges with an exponent of almost twice the magnitude. In practice so many of consecutive penalization will be too computationally expensive and they are in fact not required. Applying penalization once per time step will correct the effect of advection and diffusion to enforce a no-slip vorticity layer unless forceful acceleration or strong deformation is applied to the solid. To get the correct vorticity to enforce the no-slip condition for an onset flow consecutive penalization could be feasible at the first time step. However, the present simulations have proven this initial correction to be insignificant for the ensuing part of the simulation. And as Fig. 3.3c shows the interior velocity decreases similarly towards $\boldsymbol{u}_s$ with respect to the number of penalization repetitions independent of the penalizations being carried out in one time step or once per time step.

3.2.1 Aerodynamic forces

The two-dimensional (2D) force per unit length applied by the solid to the fluid and vice versa can be computed by integrating the penalization term over the domain. For the explicit evaluation of the penalization term the force is given by

$$\boldsymbol{F}_s = -\rho \int_A \lambda\chi(\boldsymbol{u}_s - \boldsymbol{u})\mathrm{d}A \tag{3.2.7}$$

or, in the case of split-step interpolation penalization, by integrating the time averaged rate of change of the interpolation penalization

$$\boldsymbol{F}_s = -\rho \int_A \chi\frac{\boldsymbol{u}_s^n - \boldsymbol{u}^n}{\Delta t}\mathrm{d}A. \tag{3.2.8}$$

Alternatively the force on stationary objects can be calculated by the rate of change of the integral of vorticity moments [115] over the domain A

$$\boldsymbol{F}_s = -\rho\frac{d}{dt}\int_A \boldsymbol{x}_m \times \omega_m \boldsymbol{e}_z \mathrm{d}A \tag{3.2.9}$$

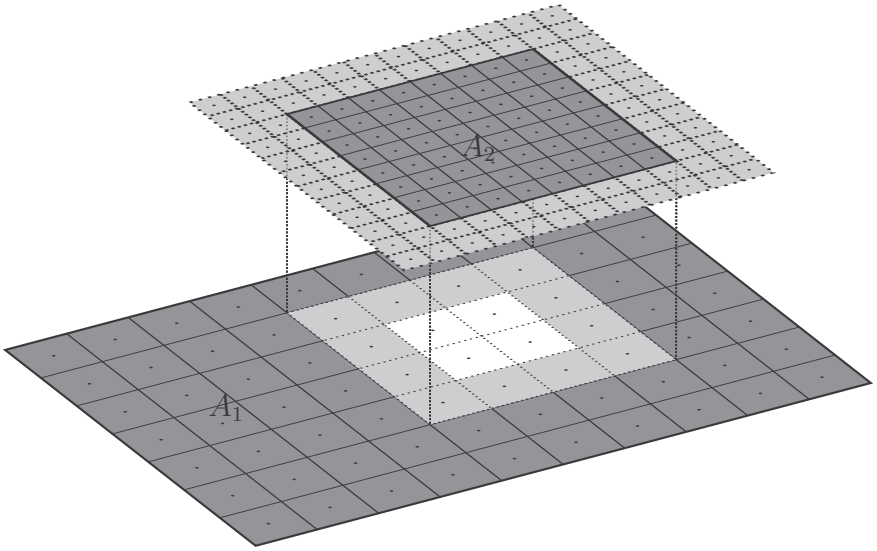


Figure 3.4: Meshes: The dark areas are the patch interior mesh A_2 and the parent mesh A_1, the light grey areas are the corresponding buffers. The cell centres have been marked by dots.

where ρ is the fluid density. The expression can be expanded to domains with a flux of vorticity across the boundaries [71] and to allow general solid body motion of the objects [108].

3.3 Multiresolution vortex-in-cell

Inherently, the Vortex-In-Cell algorithm is limited by the required uniform mesh resolution. Particles of different size can be initialised to increase the spatial resolution where needed but to be able to use FFTs to efficiently solve the Poisson equation, the underlying mesh must be uniform thus losing the added spatial resolution when computing particle velocities and when remeshing. By utilising the linearity of the Poisson equation (Eq. (3.1.4)) it is possible to increase the local resolution on patches, i.e. to have local meshes of higher resolution conforming to the requirements of the FFT.

$\bar{\omega}$ defines the complete vorticity field and $\bar{\Psi}$ and $\bar{\boldsymbol{u}}$ the corresponding stream function and velocity field on the domain A. A is composed into A_1 and A_2 where

$$A = A_1 \cup A_2$$
$$A_1 \cap A_2 = \emptyset$$

Similarly the complete vorticity $\bar{\omega}$ is divided into a part ω_1 and ω_2 contained in A_1 and A_2 respectively such that

$$\bar{\omega} = \omega_1 + \omega_2.$$

Due to the linearity of the Poisson equation Eq. (3.1.4) it follows that

$$\bar{\Psi} = \Psi_1 + \Psi_2 \tag{3.3.1}$$

where

$$\begin{aligned}\boldsymbol{\nabla}^2\Psi_1 &= -\omega_1\\ \boldsymbol{\nabla}^2\Psi_2 &= -\omega_2,\end{aligned}$$

are the stream functions on the domain A corresponding to the vorticity ω_1 and ω_2. Due to the linearity of the curl operator it follows that

$$\bar{\boldsymbol{u}} = \boldsymbol{u}_1 + \boldsymbol{u}_2, \tag{3.3.2}$$

where

$$\begin{aligned}\boldsymbol{u}_1 &= \nabla \times \Psi_1\\ \boldsymbol{u}_2 &= \nabla \times \Psi_2\end{aligned}$$

The numerical representation of ω_1 and ω_2 is contained on rectangular uniform Cartesian meshes M_1' and M_2'' covering at least A_1 and A_2 respectively. The primes $'$ and $''$ indicate mesh spacings Δx_1 and Δx_2. If $\Delta x_1 \neq \Delta x_2$ Eqs. (3.3.1) and (3.3.2) are evaluated using mesh-to-mesh interpolation using Eq. (3.1.7), where $(\omega_p,\boldsymbol{x}_p)$ is replaced by the mesh points and corresponding vorticity values. ω'' denotes vorticity on a mesh with a resolution higher than that of a mesh containing ω'. In the present work the resolution of ω'' is twice that of ω', i.e. $\Delta x_2 = \Delta x_1/2$. Similarly $\boldsymbol{u}''$ denotes velocity at twice the resolution of $\boldsymbol{u}'$.

The points of the 2D meshes are interpreted as the centres of equally sized rectangular cells. It is required that the boundaries of A_1 and A_2 are aligned with the cell sides to ensure a unique representation of the vorticity ω_1 and ω_2 and the conservation of the total vorticity, i.e. circulation. This implies that the mesh points of the two meshes are staggered in each coordinate-direction by $\Delta x_2/2$. A_1 and A_2 are shown as dark areas on Fig. 3.4.

It is possible to calculate the velocity field $\boldsymbol{u}$ on both the coarse M_1' and the finer M_2'' taking advantage of the increased resolution on M_2''. The ensemble of the mesh M_2 and the set of particles created from it is referred to as a *patch* and M_1 and the set of particles created from M_1 is called the

parent. The patch will be specified further in section 3.4. By interpolating ω_2'' to M_1' the mesh contains a complete representation of the vorticity field

$$\bar{\omega}_1' = \omega_1' + \omega_2' \tag{3.3.3}$$

and the velocity $\bar{\boldsymbol{u}}_1'$ can be determined using the standard free-space Green's function approach on the uniform M_1'. The bar denotes that the velocity field corresponds to the complete vorticity and the lower index indicates the resolution level.

For M_2'' the velocity contribution of both ω_1 and ω_2 must be included. To obtain the velocity contribution $\boldsymbol{u}_1'$ from the exterior vorticity ω_1, the velocity contribution $\boldsymbol{u}_2'$ of the coarse patch vorticity ω_2' is subtracted from $\bar{\boldsymbol{u}}_1'$. The result is then interpolated to M_2''

$$(\boldsymbol{u}_1')'' = (\bar{\boldsymbol{u}}_1' - \boldsymbol{u}_2')'' = (\bar{\boldsymbol{u}}_1')'' - (\boldsymbol{u}_2')''$$

yielding the contribution of the exterior vorticity on M_2''. The final complete velocity is given by adding $\boldsymbol{u}_2''$, the velocity contribution of ω_2'', to $(\boldsymbol{u}_1')''$

$$\bar{\boldsymbol{u}}_2'' = (\boldsymbol{u}_1')'' + \boldsymbol{u}_2'' = (\bar{\boldsymbol{u}}_1')'' - (\boldsymbol{u}_2')'' + \boldsymbol{u}_2''. \tag{3.3.4}$$

Again the bar denotes that the velocity corresponds to the complete vorticity. If multiple patches share the same parent mesh M_1 the velocity field $\bar{\boldsymbol{u}}_1'$ can be reused. This is not the case if $\boldsymbol{u}_1'$ is computed directly.

3.4 Patches and interpolation

The present approach builds on the work of [6] and thereby differs from other multiresolution particle methods as the set of all particles is divided into subsets, each subset being associated with a certain mesh. In the present work nodes of meshes of varying resolution are not aligned as [6] but staggered to ensure conservation of vorticity. The combination of a mesh and the corresponding set of particles is referred to as a patch. Each patch has a parent; a mesh and its associated set of particles, and the entire data structure consists of patches. The parent can be another patch of any resolution level or it may be the lowest level mesh and its associated particles. The mesh that covers the entire computational domain and its associated particles are level 1 and are referred to as the base patch. As the base patch does not have a parent patch, parts of the interpolation and velocity computation vary for the base patch. The resolution of the l-th level patch is given relative to the base mesh spacing Δx_1. Thus

$$\Delta x_l = \Delta x_1 \left(\frac{1}{2}\right)^{l-1}$$

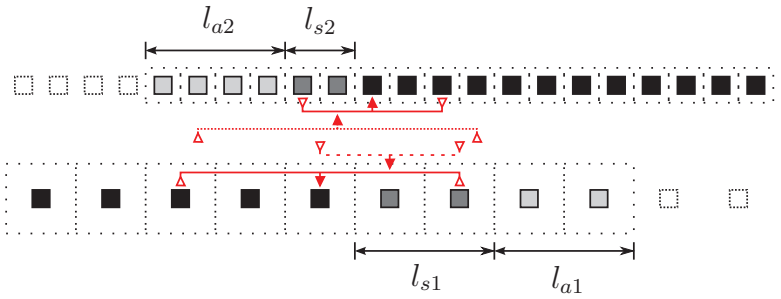


Figure 3.5: Sketch of the patch/parent staggered meshes at patch boundary. The dashed lines represent the volumes corresponding to the mesh points. The solid black squares are the unique patch interior mesh points and unique points of the parent mesh. Solid dark/light grey points are support/advection buffer points. Solid arrows indicate the required support of the interpolation kernel and dashed/dotted arrows indicate the required support for interpolating the patch/-parent particles to the patch/parent buffers.

for any $l > 0$. In the present implementation a patch of level $l > 1$ has a parent of level $l-1$. By extending the interpolation, patches could have parents of even coarser resolution. For each level there may be several patches with the same or different parents and P_l^p denotes the p-th patch of level l. To ensure spatial support for the patch particle-to-mesh interpolation (P2M) (Eq. (3.1.7)), particles are formed from both the *interior* patch as well as a surrounding *support* buffer [6]. k denotes the width of the support; for the M_4' kernel $k = 2$ cf. Eq. (3.1.8). These particles are created during the particle remeshing step to ensure support for the $M'4$ kernel for the interior mesh points close to the boundary. Fig. 3.5 shows the discrete support buffers for the staggered setup of the patch and the parent mesh

$$l_{s1} = k\Delta x_1 \tag{3.4.1}$$

$$l_{s2} = k\Delta x_2. \tag{3.4.2}$$

When the particles are advected after the remeshing at every n_r time steps, the interpolation support of the outermost mesh points is ensured by creating additional particles in the *advection* buffer [6] of width

$$l_{a1} = \sum_{n=1}^{n_r} ||u^n||_\infty \Delta t \tag{3.4.3}$$

$$l_{a2} = \sum_{n=1}^{n_r} ||u^n||_\infty \Delta t. \tag{3.4.4}$$

The advection buffer is truncated to the lowest multiple of the cell length. To evaluate the right hand side of Eq. (3.1.1) with second order finite differences

the buffer must be either expanded additionally by one cell for support of the finite difference scheme or, as in the present study, through one-sided finite differences for the outermost layer of cells.

The present mesh and buffer setup ensures support for the mesh values of the patch buffer such that values can be interpolated from the particles of the parent, and vice versa, as shown by the dashed and dotted arrows on Fig. 3.5. Thereby the meshes, including buffers, are populated entirely by particle-to-mesh interpolation. In two dimensions the buffers enclose the patch interior as shown on Fig. 3.4. It is brought to mind that vorticity is only considered unique in the patch interior, i.e. excluding buffers, and the part of the parent mesh that is not covered by the patch interior.

3.4.1 Mesh-to-mesh interpolation in the buffer

In the present work the value of the buffer mesh points are obtained by mesh-to-mesh (M2M) interpolation. By extending the patch support buffer both parent and patch buffer mesh values can be obtained entirely through mesh-to-mesh interpolation after particle values have been interpolated to the unique interior points. This reduces the computational overhead for the structured meshes. To ensure support for the $M'4$ kernel the patch buffer $l_{s2} + l_{a2}$ is increased by one mesh point as indicated by the dashed arrow on Fig. 3.5 and l_{a2} is truncated to nearest multiple of Δx_2. This extension of the buffer has a beneficial side effect when dealing with discontinuities at the patch boundary as described in section 3.7.1. After performing particle-to-mesh interpolation to the unique vorticity regions of the patch and parent the patch vorticity is interpolated not just to the parent buffer. Instead the entire patch vorticity is interpolated to the parent mesh corresponding to Eq. (3.3.3). Finally the patch buffer is filled by mesh-to-mesh interpolation from the parent mesh. Alternatively the parent buffer can be similarly extended but resulting in severe limitations when adjacent patches exist on the same level.

3.4.2 Upgrading neighbour buffer

Adjacent patches have an overlap of the interior and the buffer of the neighbouring patch. After particle-to-mesh interpolation the buffer vorticity is *upgraded*, i.e. copied from the neighbouring patch to the buffers, where possible. Thereby the existing buffer vorticity, stemming from the lower resolution parent mesh, is upgraded by copying the same resolution neighbour vorticity. After the velocity has been computed on all patches and levels

velocity is upgraded as well. This increases the direct connection between patches.

In case that the vorticity cannot be upgraded from neighbouring patches the coarser resolution vorticity interpolated from the parent is left in the buffer, which in turn is advected into the patch interior. The coarser incoming buffer vorticity is sufficient under the assumption that the complexity of the flow is formed on patch levels of higher resolution.

3.4.3 Multiresolution and penalization

Penalization is carried out entirely on the meshes. Since the width of the mollification interval L and in turn S and χ depend on the mesh resolution, penalization is carried out only on patches of same resolution in the present implementation. The resulting vorticity is interpolated and copied to the other levels. Alternatively L could be geometrically fixed causing the mollification interval to either span a large number of cells at the highest level or being under-resolved on the lowest level. However, the error of the finite difference approximation introduces a discrepancy of the vorticity of levels of different resolution.

The force calculated from Eq. (3.2.8) is thus only integrated on the level where penalization is performed. When calculating the force from the vorticity moments Eq. (3.2.9) only unique vorticity is included. That is, vorticity is only included once; the version of highest resolution and not the lower resolution vorticity of parents.

3.5 Vortex-In-Cell algorithm using patches

The vorticity is initialised on the patch particles. In the time integration loop the particle vorticity is interpolated to the meshes. First particle values are interpolated to their patch mesh. The patches are traversed from the highest (finest) level down to level 2, copying and interpolating the patch mesh vorticity to the parent mesh. Then, traversing the patches from level 2 to the highest level, the parent mesh vorticity is interpolated and copied to the patch buffer. The velocity on level 1 is calculated by the standard free-space VIC method. Going from level 2 to the highest level the velocity is calculated as described in section 3.3. Interpolation penalization is carried out on the levels where $\chi \neq 0$, vorticity is updated and the velocity is recalculated from the updated vorticity field. Particles are remeshed, i.e. new particles are formed at the mesh nodes, and subsequently advected and updated due to viscous diffusion. This is done by interpolating the mesh based velocities

and $\frac{D\omega}{Dt}$ to the particles. The advection and particle diffusion may be applied in one (Euler) step or by a higher order integration scheme. In the latter case the particle vorticity is interpolated to the mesh, velocities and $\frac{D\omega}{Dt}$ are calculated and interpolated back to the particle positions in each sub-step. Algorithm 1 shows this more explicitly.

If the time integration of the particle positions and vorticity is carried out using a higher order integration method the particle-to-mesh interpolation, velocity computation, vorticity equation and mesh-to-particle interpolation steps are carried out for each sub-step of the scheme.

A patch may be moved during simulation by offsetting the bounds before the particle-to-mesh interpolation. This adaptivity alleviates tracking features of the flow.

3.5.1 Implementation

The present work is implemented in `MATLAB` and therefore certain considerations must be taken. As loops and conditional statements are generally very slow the implementation has been vectorised to the extent possible. For free-space simulations, particles leaving the domain are not removed from the computation. Instead the strength is set to zero and the particle is reintroduced periodically into the computational domain.

3.6 Fourier transforming discontinuous fields

If a patch P_l^p is located across a region of finite vorticity the patch vorticity field ω_l^p will be discontinuous across its boundary ∂P_l^p. As the vorticity of P_l^p is copied to the parent, the parent vorticity field will not be discontinuous at ∂P_l^p. Fourier transforming ω_l^p into $\hat{\omega}_l^p$ introduces artificial frequencies known as the Gibbs phenomenon. As suggested in [36] discontinuities can be avoided by positioning ∂P_l^p outside regions of high vorticity. However, this severely constrains the versatility of the patches.

Alternatively one can apply spectral filtering, e.g. the σ-approximation [53], by multiplying the k-th frequency of $\hat{\omega}^p$ by the normalised sinc function

$$\frac{\sin(\pi k_i/m)}{\pi k_i/m} \tag{3.6.1}$$

in each spectral direction i, where m is the number of discrete frequencies.

If Ω^p defines the entire patch vorticity, including both the interior vorticity ω^p and the buffer vorticity, another approach is windowing Ω^p to reduce

Initialisation: Create particles on all patches P_l^p carrying the initial vorticity field
while $t < T$ **do**
- **for** l from 1 to $l_{\max}$ **do**
  - **for** p from 1 to $p_{\max}(l)$ **do**
    - Interpolate the particle vorticity to the corresponding mesh ($\mathcal{O}(N_l^p)$)
- **for** l from $l_{\max}$ to 2 **do**
  - **for** p from 1 to $p_{\max}(l)$ **do**
    - Interpolate and copy the mesh vorticity to the parent mesh (M2M) hereby also filling the buffer ($\mathcal{O}(N_l^p)$)
- **for** l from 2 to $l_{\max}$ **do**
  - **for** p from 1 to $p_{\max}(l)$ **do**
    - Interpolate and copy the parent mesh vorticity to the patch buffer (M2M) ($\mathcal{O}(N_l^p)$)
- Upgrade the buffer vorticity for adjacent patches. The vorticity is completely defined on all meshes
- **Velocity computation**
- On P_1^1 calculate the velocity $\bar{\boldsymbol{u}}_1^1$ by standard free-space VIC ($\mathcal{O}(N_l^p log(N_l^p))$)
- **for** l from 2 to $l_{\max}$ **do**
  - **for** p from 1 to $p_{\max}(l)$ **do**
    - Interpolate the coarser velocity $\bar{\boldsymbol{u}}_{l-1}^d$ to M_l^p where P_{l-1}^d is the parent of patch P_l^p ($\mathcal{O}(N_l^p)$)
    - Interpolate $\omega_l^{p\prime\prime}$ to $\omega_l^{p\prime}$ ($\mathcal{O}(N_l^p)$)
    - Compute $\boldsymbol{u}_l^{p\prime}$ and $\boldsymbol{u}_l^{p\prime\prime}$ ($\mathcal{O}(N_l^p log(N_l^p))$)
    - Compute $\bar{\boldsymbol{u}}_l^p$ cf. Eq. (3.3.4) ($\mathcal{O}(N_l^p)$)
- Upgrade the buffer velocity for adjacent patches
- **Penalization**
- **for** l from 1 to $l_{\max}$ **do**
  - **for** p from 1 to $p_{\max}(l)$ **do**
    - Impose interpolation penalization cf. Eqs. (3.2.4) and (3.2.5) on the relevant levels ($\mathcal{O}(N_l^p)$)
- **for** l from $l_{\max}$ to 1 **do**
  - **for** p from 1 to $p_{\max}(l)$ **do**
    - Interpolate and copy the updated vorticity to the parent meshes ($\mathcal{O}(N_l^p)$)
    - Create new remeshed particles from the penalized vorticity ($\mathcal{O}(N_l^p)$)
    - Recalculate the velocity from vorticity ($\mathcal{O}(N_l^p log(N_l^p))$)
- **Vorticity equation**
- **for** l from 1 to $l_{\max}$ **do**
  - **for** p from 1 to $p_{\max}(l)$ **do**
    - Compute $\frac{D\omega}{Dt}$ on the mesh cf. Eq. (3.1.1) ($\mathcal{O}(N_l^p)$)
- **Mesh-to-particle interpolation**
- **for** l from 1 to $l_{\max}$ **do**
  - **for** p from 1 to $p_{\max}(l)$ **do**
    - Interpolate the velocity and $\frac{D\omega}{Dt}$ to the particles ($\mathcal{O}(N_l^p)$)
- **Particle position and vorticity update**
- **for** l from 1 to $l_{\max}$ **do**
  - **for** p from 1 to $p_{\max}(l)$ **do**
    - Update the particle positions and vorticity values ($\mathcal{O}(N_l^p)$)

Algorithm 1: The multiresolution VIC algorithm using patches.

discontinuities of the vorticity

$$\tilde{\Omega}^p(\boldsymbol{x}) = \Omega^p(\boldsymbol{x})w(\boldsymbol{x}), \tag{3.6.2}$$

where $w(\boldsymbol{x}) \in [0;1]$ is the window function. No additional interpolation is required as the buffers are filled in the particle-to-mesh interpolation process. The buffers either contain vorticity interpolated from the coarse parent mesh or copied from a same resolution neighbour patch. Therefore Eq. (3.3.4) is calculated on the entire patch, buffers included, but the result is only applied to the patch interior. In the buffer $\bar{\boldsymbol{u}}'$ is used. If a neighbour patch exists the buffer velocities are updated from the neighbour interior. Therefore discontinuities only pose a problem for the patch interior. The correct velocity field is computed in spite of the additional buffer vorticity in the velocity computations due to the linearity of Eq. (3.1.4). That is, in Eq. (3.3.4) the buffer vorticity adds a contribution to $\boldsymbol{u}_2''$ which is in turn subtracted in $(\boldsymbol{u}_2')''$.

The maximum error due to the spectral and geometrical truncation of the numerical evaluation of the Fourier transform converges spectrally to machine precision by constraining the shape of w [94]. Notably the differentiability of w. An exponential window function is proposed [94]

$$w_{\exp}(x) = 10^{-a^n|2(x-x_L)/(x_R-x_L)-1|^n} \tag{3.6.3}$$

where x_l,x_r and x_L,x_R are the left/right bounds of the patch and the buffer regions respectively as sketched on Fig. 3.6a. The parameters $a = 1.111, n = 27$ have been found such that the error in the patch interior converges spectrally on the order of machine precision given that the buffers on both sides are $\approx 1.8|x_r - x_l|$ wide [94]. For such an extensive buffer $w_{\exp}$ ensures minimal error and spectral convergence but due to the requirements to the buffer width other windows functions have been investigated. Furthermore the spectral convergence is not of importance for the present work as derivatives are found using a second order finite difference approximation.

The following window functions are 0 outside the interval $x \in [x_L; x_R]$. The Heaviside window function

$$w_H(x) = 1, \quad x_L \leq x \leq x_R \tag{3.6.4}$$

corresponds to moving the discontinuity away from the patch interior by extending the vorticity to the buffer. The Tukey window [41] ramps the the

buffer vorticity sinusoidally to zero and is defined as

$$w_T(x) = \begin{cases} \frac{1}{2}\left(1 - \cos\left(\pi\frac{x - x_L}{x_l - x_L}\right)\right) & , \quad x_L < x < x_l \\ 1 & , \quad x_l \leq x \leq x_r \\ \frac{1}{2}\left(1 - \cos\left(\pi\frac{x_r - x}{x_R - x_r}\right)\right) & , \quad x_r < x < x_R \end{cases} \tag{3.6.5}$$

which is first order continuous. The polynomial window function

$$w_{P^\rho}(x) = \begin{cases} 1 - \left(\frac{x_l - x}{x_l - x_L}\right)^\rho & , \quad x_L < x < x_l \\ 1 & , \quad x_l \leq x \leq x_r \\ 1 - \left(\frac{x - x_r}{x_R - x_r}\right)^\rho & , \quad x_r < x < x_R \end{cases} \tag{3.6.6}$$

is order ρ continuous. An error is introduced to Eq. (3.3.4) unless the windowing of the vorticity is conservative when applied at staggered meshes of different resolutions. To reduce this error the non-linear window functions $w_{\exp}$, w_T and w_{P^ρ} are modified such that, at the coarse mesh the window function is evaluated as

$$w(x_i') = \frac{w(x_i' - \Delta x_1/4) + w(x_i' + \Delta x_1/4)}{2}. \tag{3.6.7}$$

In the following analysis this has lowered the error by one or two orders of magnitude.

3.6.1 Quantitative study of the discontinuities

The window functions Fig. 3.6a and the σ-approximation lead to different errors. To compare the window functions a one-dimensional study of the multiresolution procedure is carried out. A mesh is defined from $x = 0$ to $x = l_1$ with mesh spacing Δx_1. A patch of length l_2 is defined from $x_l = 5/12 l_1$ to $x_r = 7/12 x_l$ with buffers varying in width from a few cells to $1.8 l_2$. The buffers are defined from x_L to x_l and from x_r to x_R. The sinusoidal vorticity

$$\omega(x) = \sin\left(\frac{2\pi x}{l_1}\right), \tag{3.6.8}$$

see Fig 3.6b, is initialised on all discrete meshes including buffers and the fine and coarse patch meshes.

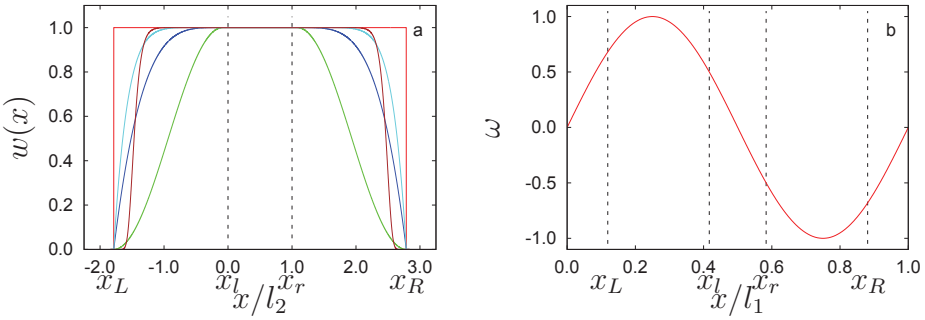


Figure 3.6: (a) The window functions on the buffer around the patch interior (vertical lines). The Heaviside (red), Tukey (green), P^4 (blue), P^6 (cyan) and exponential (brown) window functions on a very wide buffer patch. (b) The one-dimensional vorticity on the domain $[0; l_1]$. The vertical lines mark the patch interior and buffers.

On the entire patch the vorticity Ω_2' and Ω_2'' are initialised with mesh spacings of Δx_1 and $\Delta x_2 = \Delta x_1/2$ respectively. On the parent mesh the complete vorticity $\bar{\omega}_1'$ is initialised. In the case of spectral filtering the buffer is set to 0 otherwise a window function is applied to the patch vorticity. The stream functions $\bar{\Psi}_1'$, Ψ_2', Ψ_2'' and the velocities $\bar{u}_1'$,u_2',u_2'' corresponding to $\bar{\omega}_1'$,ω_2',ω_2'' are computed via the free-space FFT method using the 1D Green's function [4]

$$G_{1D} = \frac{1}{2}|y - x| \tag{3.6.9}$$

and $\mathcal{O}(\Delta x^2)$ finite differences. The multiresolution velocity

$$\bar{u}_2'' = (\bar{u}_1')'' - (u_2')'' + u_2'' \tag{3.6.10}$$

is computed using the $M'4$ interpolation kernel. The solution is compared to the free-space velocity $\bar{u}_1''$ corresponding to the continuous smooth complete domain vorticity $\bar{\omega}_1''$ with uniform mesh resolution $\Delta x_1/2$. For several window functions the error

$$\epsilon_v = \bar{u}_1'' - \bar{u}_2'' \tag{3.6.11}$$

has been investigated in and around the patch buffer region and the patch interior. The error ϵ_v stems from Fourier transforming the discontinuous vorticity fields as well as the interpolation between meshes, finite difference evaluations and FFT of the first order continuous Green's function.

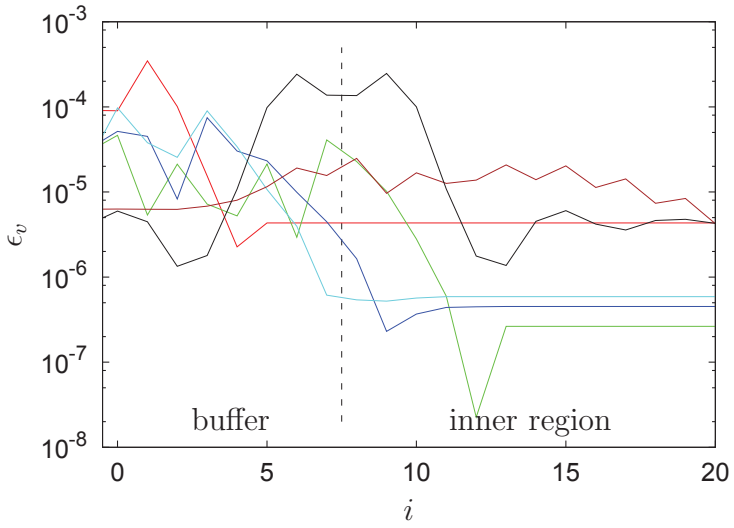


Figure 3.7: (a) Magnitude of the patch velocity error ϵ_v with the w_H (red), w_T (green), w_{P^4} (blue), w_{P^6} (cyan), $w_{\exp}$ (brown) masks and the σ-correction (black) as a function of mesh index i. The buffer is $8\Delta x_2$ wide and $i = 0$ at the outermost buffer point.

Fig. 3.7 shows the magnitude of ϵ_v for a buffer of width $8\Delta x_2$ using different window functions and filters. The error fluctuates around $\epsilon_v = 0$ mainly when applying w_H or the σ-approximation but only in the latter case does the error assume both positive and negative values in the patch interior.

The Heaviside window w_H corresponds to extending the vorticity into the buffer and moving the discontinuity away from the region of interest; the patch interior. w_H is independent of the buffer width and produces the error of greatest magnitude. However, only $6\Delta x_2$ mesh points inside the outer buffer bounds the deviation to the reference velocity field is both of low magnitude and is smooth. This does, in most cases, not pose an additional constraint to the buffer width due to the requirements of the interpolation and advection buffers, Eqs. (3.4.2) and (3.4.4). Since vorticity is already interpolated to the buffer region, including the buffer vorticity in the velocity computations has minimal impact on the algorithmic complexity. Applying the σ-approximation to the unwindowed vorticity field only decreases the error by 30%. Not only does the magnitude of the error remain substantial but the velocity oscillations introduce artificial vorticity when calculating the corresponding vorticity Eq. (3.2.3).

The polynomial windows w_{P^ρ} approach w_H in the limit of $\rho \to \infty$ and, as w_H, they move the discontinuity away from the interior patch while being

$\mathcal{O}(\rho)$ continuous in the ramping region. w_{P^ρ} lowers and distributes the error in the buffer but, depending on the value of ρ, some fluctuations persist in the patch interior. As shown in Fig. 3.7 the constant level error increases with ρ while moving the fluctuations away from the interior. This indicates that an optimal relation exists between ρ and the number of mesh points in the buffer. This has not been pursued in the present work.

The $w_{\exp}$ window is not designed for the current buffer setup and as expected ϵ_v remains substantial far into the interior patch when applying $w_{\exp}$. At the centre of the patch ϵ_v assumes the minimum value of for all window functions (not shown). Applying the w_T window leads to a lower and more uniform error in the buffer than is the case for w_H but it extends into the patch interior by a fixed number of mesh points. Its minimum value equals that of the $w_{\exp}$ window and the present results indicate that using w_T in part of the buffer may be optimal.

3.7 Results

The presented multiresolution velocity algorithm is validated against the Perlman vorticity patch [75]. To validate the interpolation penalization, the impulsively started flow around a circular cylinder is simulated. The drag and lifting forces are investigated for both static and rotating cylinders. Also the flow around the Great Belt East suspension bridge deck is simulated and compared to solutions of other numerical methods.

3.7.1 Perlman vorticity patch

To test the calculation of the fluid velocity separately the radially symmetric patch of vorticity is considered as defined by [75]

$$\omega(r) = \begin{cases} \left(1-r^2\right)^7 & , \quad r \leq 1 \\ 0 & , \quad r > 1 \end{cases} \tag{3.7.1}$$

where $r = \sqrt{x^2+y^2}$ is the radial distance from the centre of the domain, and

$$\boldsymbol{u}_p(x,y) = \begin{cases} -\dfrac{1}{16r^2}\left(1-\left(1-r^2\right)^8\right)\begin{pmatrix} y \\ -x \end{pmatrix} & , \quad r \leq 1 \\ -\dfrac{1}{16r^2}\begin{pmatrix} y \\ -x \end{pmatrix} & , \quad r > 1 \end{cases} \tag{3.7.2}$$

is the corresponding velocity field. Fig. 3.8a shows the initialised vorticity field and the patches. The extend of the computational domain is 3×3 and

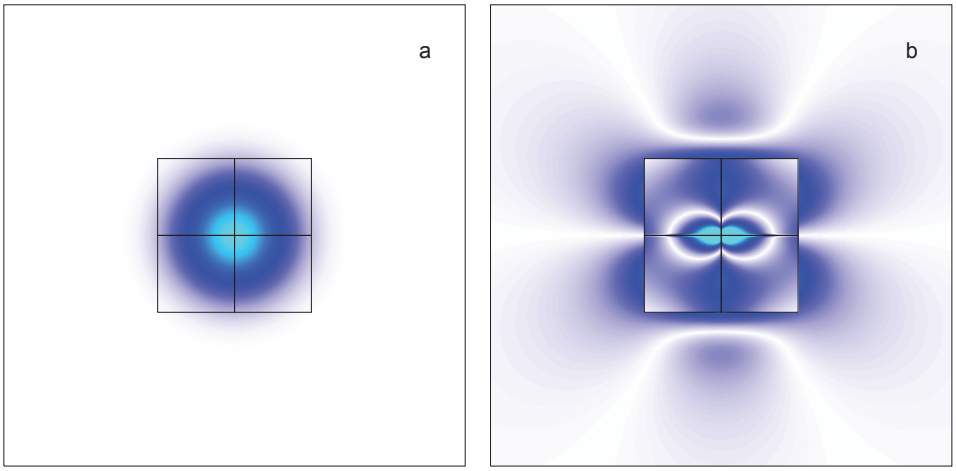


Figure 3.8: (a) The Perlman patch vorticity field where white, blue, cyan correspond to $|\omega| = 0.0, 0.5, 1.0$. (b) The error on the horizontal velocity component where white, blue, cyan corresponds to $|\epsilon_x|/|\epsilon_{\max}| = 0.0, 0.5, 1.0$. The black lines indicate the location of the patches.

the 1×1 centre of the domain is divided into four equally sized quadratic patches with side length $l_2 = 1/2$. Thus the patch vorticity field is discontinuous at the patch boundary which may lead to Gibbs' phenomenon when performing the Fourier transform. As discussed the Gibbs' oscillations can be suppressed by windowing [41] and spectral convergence can be achieved [94] by careful selection of windowing functions and by extending the size of the buffer. However, this requires a 360% increase of the domain size in each physical direction and since $\mathcal{O}(2)$ finite differences are applied to the result spectral convergence is not of interest. It has been found that a convergent solution can be obtained without windowing. Thus, by extending the vorticity field into the buffer the discontinuity and associated errors are moved away from the region of interest; the patch interior. Velocity values in the buffer are then filled from the parent or neighbouring patches.

Fig. 3.8b shows the error to the velocity solution

$$\boldsymbol{\epsilon} = \boldsymbol{u}_p - \boldsymbol{u} \tag{3.7.3}$$

on the domain. Each patches is resolved by 256×256 cells and has a buffer of 8 cells. A significant error is caused at this relatively thin buffer width. Fig. 3.9a shows the integral absolute error of the numerical result to Eq. (3.7.2) over the patch interior as function of the number mesh cells N along a side of the quadratic patches. For buffer widths of $1.8l_2$ the integral error converges according to the $\mathcal{O}(\Delta x^2)$ finite difference scheme used. For

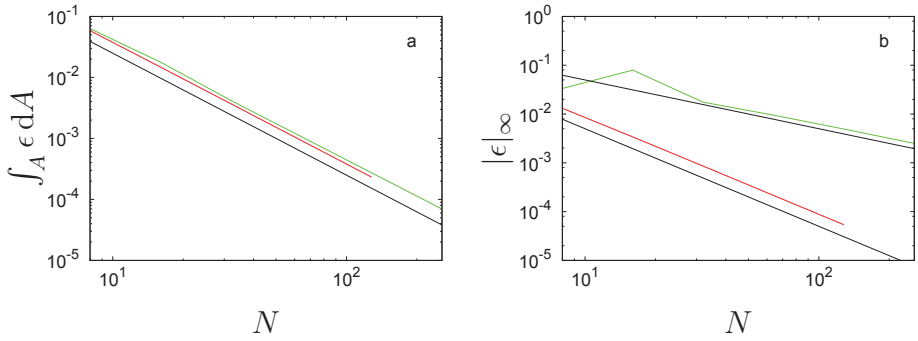


Figure 3.9: Spatial convergence study for the Perlman vorticity patch Eq. (3.7.1) while extending the vorticity field to a $1.8l_2$ buffer (red) or a $0.18l_2$ buffer (green): (a) Integral of the magnitude of the deviation of the computed velocity to Eq. (3.7.2) as a function of the number of mesh cells N along the side of the quadratic patch. (b) The maximum patch interior error as a function of N.

a buffer width of $0.18l_2$, the integral error is increased by 20–40 % but converges $\mathcal{O}(\Delta x^2)$. Fig. 3.9b shows the maximum error. For the buffer width of $1.8l_2$ the maximum error is aligned with the N^{-2} trend line, the expected convergence of the $\mathcal{O}(\Delta x^2)$ the finite difference approximation. However, by decreasing the buffer width to $0.18l_2$ the error converges $\mathcal{O}(\Delta x)$, indicating that errors due to discontinuities do not dominate the $1.8l_2$ errors.

On the coarsest mesh it may be necessary to mollify the vorticity field at the domain boundary by modifying and reducing the vorticity field. As proposed in [89] the boundary vorticity is ramped sinusoidally to zero in the present work. Compared to the vorticity being discarded as it is advected across the domain boundary this is an insignificant modification of the vorticity field.

3.7.2 Impulsively started flow past a circular cylinder at Re = 550

The impulsively started flow around a circular cylinder has been thoroughly studied in the literature e.g. [52, 89]. The flow is simulated at Reynolds number

$$\mathrm{Re} = \frac{U2R}{\nu}$$

$\mathrm{Re} = 550$, where U is the far-field velocity and R is the radius of the cylinder. Initially the domain is resolved by one uniform patch, i.e. by the standard VIC algorithm. The mesh spacing is $\Delta x_1 = R/256$ and the cylinder is

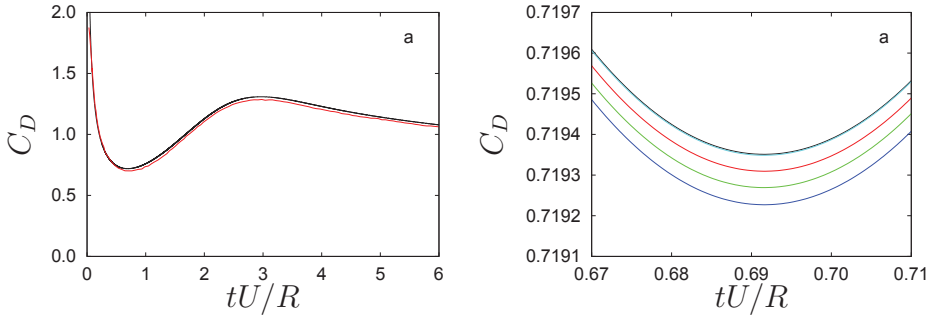


Figure 3.10: (a) Coefficient of drag for onset flow around circular cylinder. The present results (red) agree with the results of Koumoutsakos and Leonard [52] (black) within 3 % for Re = 550. (b) Close-up of C_D for uniform resolution by penalization term (black) and vorticity moments (green). C_D for the quartered patch layout by penalization term (red) and vorticity moments (blue). C_D for a boundary and wake patched domain by vorticity moments (cyan).

centred at $[2R, 2.5R]$ from the bottom left corner of the $8R \times 5R$ domain. The results are presented in non-dimensional time tU/R and the dimensionless time step size is $\Delta tU/R = 2.5 \cdot 10^{-4}$. Simulation is carried out until $tU/R = 6$. The circular cylinder is imposed by the interpolation-penalization method and is defined by χ and S as described in section 3.2 with $L = 3\sqrt{2}\Delta x_1$. The force acting on the cylinder is calculated by integrating the time averaged change of the penalization Eq. (3.2.8) and from the vorticity moments Eq. (3.2.9).

Fig. 3.10a shows the time evolution of drag coefficient

$$C_D = \frac{F_D}{\frac{1}{2}\rho U^2 D}, \tag{3.7.4}$$

where F_D is the drag force; the streamwise component of the aerodynamic force vector. The agreement with the results of Koumoutsakos and Leonard [52] is excellent until $tU/R = 0.5$ where the present results deviate by assuming a slightly higher value of C_D. This offset remains of the same order of magnitude and within 3 % of the reference. It is conjectured that the offset may be due to the proximate definition of the solid mask χ.

To investigate the effect of the present multiresolution algorithm the resolution of the base patch is reduced to $\Delta x_1 = R/128$ and the base patch is filled entirely by patches of resolution $\Delta x_2 = R/256$. The patch buffers are $18\Delta x_2$ wide. This *quartered* layout of patches can be seen on Fig. 3.11. By only penalizing the flow at level 2 the solution is comparable to that of the uniformly resolved domain. Fig. 3.10b shows a close up of the local minimum

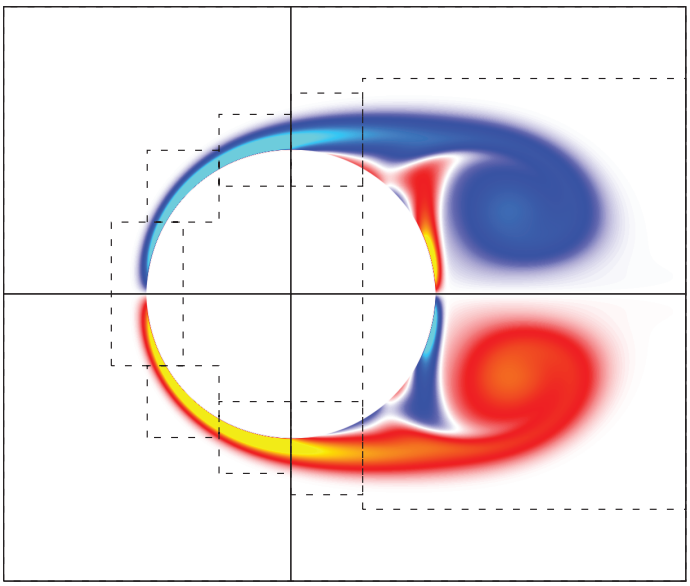


Figure 3.11: Snapshot of vorticity for the impulsively started flow around a cylinder at Re = 550 and $tU/R = 4$. The domain is overlaid by the quartered layout of patches (———) and the 8 boundary and wake patches (- - - - - -).

of C_D at $tU/R \approx 0.7$. The simulation is carried out until $tU/R = 1.25$. The coefficient of drag is 0.02 % higher for the quartered patch layout than for the uniform reference solution. For the uniform domain and the quartered patch layout C_D calculated with vorticity moments is 0.011 % and 0.016 % lower than when integrating the penalization term.

By adding patches at the boundary of the cylinder and in the immediate wake, see Fig. 3.11, the computational resources are focused more efficiently at the high vorticity regions. As for the quartered layout the patch buffers are $18\Delta x_2$ wide. The varying spatial resolution, in particular in the regions of lower resolution, could potentially alter the simulation. However, the most significant impact of this setup is the absence of patches in the solid interior. Because the velocity by which the aerodynamic forces are calculated (Eq. (3.2.8)) is found from the vorticity field diffused by Eq. (3.1.1) some finite deviation to the solid velocity field must be expected in the interior of the solid. Due to the numerical evaluation of the curl (Eq. (3.2.3)) at different resolutions, the resulting vorticity and in turn the corresponding velocity will vary when evaluating Eq. (3.2.8). When the force is partially evaluated on the coarser mesh in the solid interior this results in a 13 % deviation for the current setup. The vorticity moments depend only on the evolution of the vorticity and give correct results as shown in Fig. 3.10b.

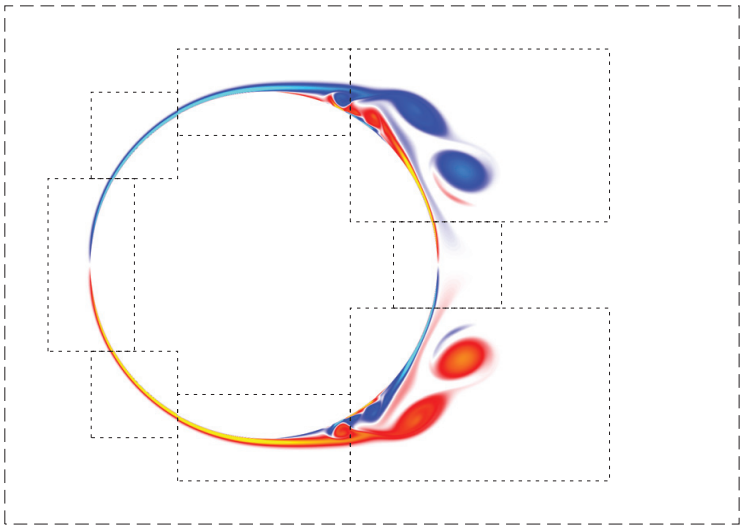


Figure 3.12: The vorticity around a cylinder in an onset flow for Re = 9500 at $tU/R = 3$. The lines (– – – –) and (·····) show the patches of resolution $\Delta x_2 = R/256$ and $\Delta x_3 = R/512$. The base patch (level 1) is not shown.

Therefore for more complex arrangements of patches the vorticity moments should be used to compute the body forces.

3.7.3 Impulsively started flow past a circular cylinder at Re = 9500

Increasing the Reynolds number to 9500 leads to a more complex flow requiring a finer resolution and time integration, i.e. $\Delta tU/R = 10^{-4}$. Fig. 3.12 shows the vorticity field at $tU/R = 3.0$ overlaid by patches of varying resolution. On the base mesh one patch of twice the base mesh resolution ($\Delta x_2 = R/256$) is positioned around the cylinder and its wake. This patch has a buffer of width $8\Delta x_2$. On this patch several patches of four times the base mesh resolution ($\Delta x_3 = R/512$) is positioned on the cylinder boundary and the immediate wake. All level 3 patches have buffers $18\Delta x_3$ wide. The drag force shown in Fig. 3.13 is computed using vorticity moments and matches previous results cf. [52, 89]. More visualisations of the flow can be found on Fig. G.1 in Appendix G.

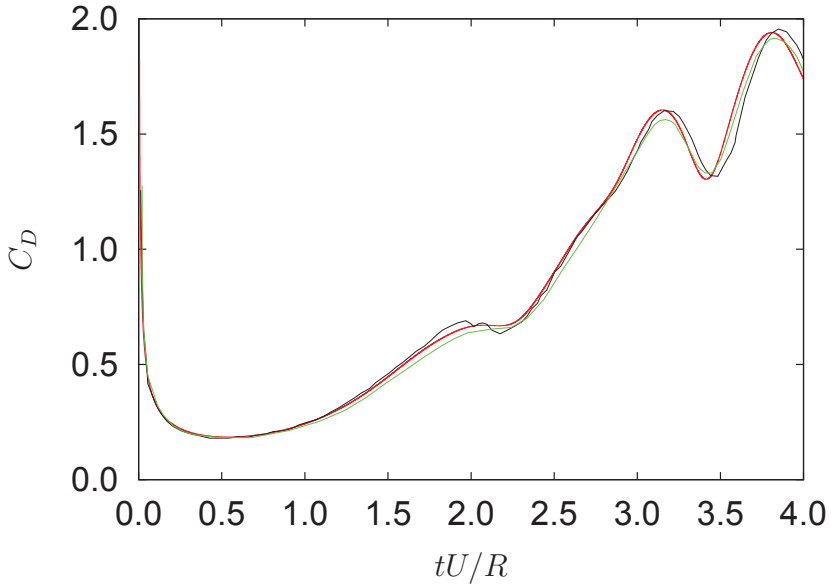


Figure 3.13: The calculated C_D (red) for the onset flow around a circular cylinder at Re = 9500 shows good agreement with the results of Koumoutsakos and Leonard [52] (black) and Rossinelli et al. [89] (green).

3.7.4 Impulsively started flow around a rotating circular cylinder at Re = 1000

A cylinder is centred in the y-direction in a $16R \times 10R$ domain at a $\Delta x_1 = R/64$ resolution. Two nested patches cover the cylinder and its immediate wake with a resolution $\Delta x_3 = R/256$ at the highest level (not shown). The buffers are of width $6\Delta x_2$ and $10\Delta x_3$ respectively. The solid body rotation velocity $\boldsymbol{u}_s$ is initialised with the ratio of the counter-clockwise circumferential velocity to the far-field velocity $R\Omega_s/U = 1$, where Ω_s is the angular velocity of the cylinder. Fig. 3.14 shows the computed coefficient of drag and coefficient of lift $C_L = F_L/(\frac{1}{2}\rho U^2 D)$, where F_L is the component of the aerodynamic force normal to the direction of the flow, i.e. the lifting force. The drag deviates by 6.5 % to the results of Chou [14] and apart from the initial development C_L deviates by 2.5 % from this reference solution. The simulations has run at two different resolutions with insignificant variation of the results.

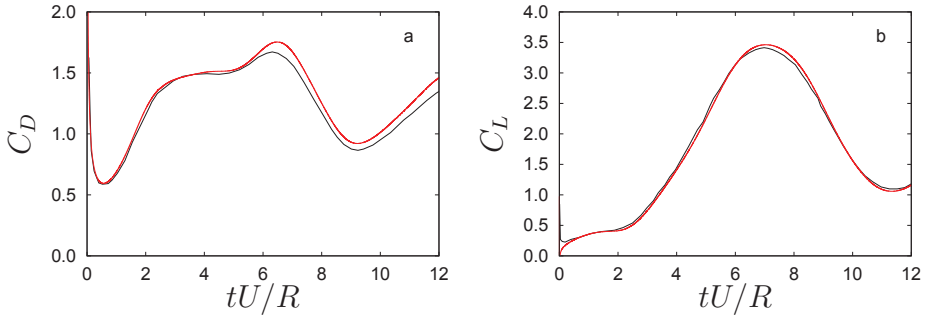


Figure 3.14: (a) C_D and (b) C_L for the impulsively started flow around a cylinder rotating with a non-dimensional speed of $R\Omega_s/U = 1$. Present values (red) and the reference by Chou [14] (black).

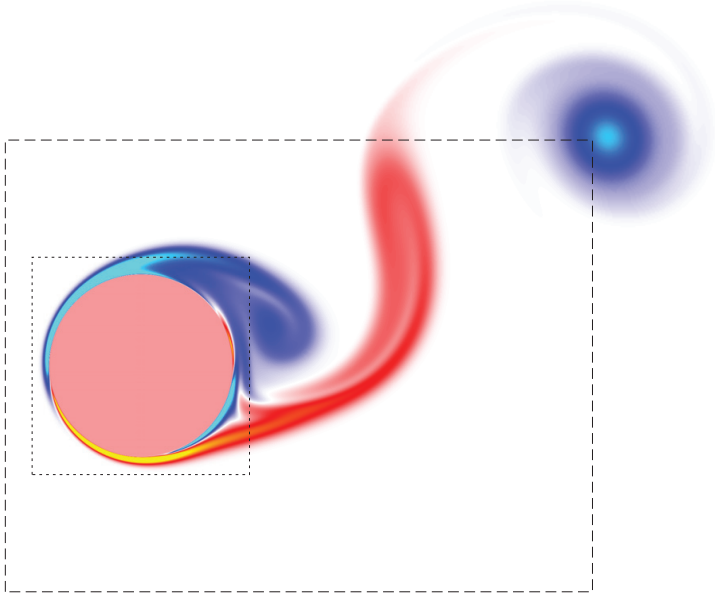


Figure 3.15: The vorticity around a rotating cylinder at $tU/R = 11$ with $R\Omega_s/U = 1$ and Re $= 1000$. The lines (– – – –) and (·····) show the patches of resolution $\Delta x_2 = R/128$ and $\Delta x_3 = R/256$. The base patch (level 1) is not shown.

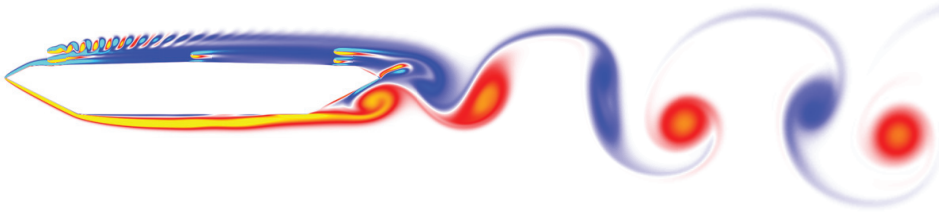


Figure 3.16: The vorticity field around the Great Belt East suspension bridge deck at $tU/C = 6.9$.

3.7.5 Flow around Great Belt East suspension bridge deck

To demonstrate the versatility of the method the flow around the Great Belt East bridge is simulated at $\mathrm{Re} = 10000$ based on the chord C and the free-stream velocity. The geometry includes railing and the crash barrier cf. Fig. 3.16, which require a high resolution to resolve properly. This, in turn, requires a low time step to ensure stability of the diffusion term. The measured time average of the lift and drag, $C_L = -0.0445$ and $C_D = 0.0533$, deviates by 11.6 % and 3.08 % respectively compared to the results of the flow solver `DVMFLOW` [108, 64]. The vorticity field (Fig. 3.16) shows large scale vortex shedding in the wake, but also higher frequency vortex shedding from the upstream railings.

3.8 Summary

The introduced novel multiresolution VIC algorithm is based on free-space FFTs for the solution of the Poisson equation on the patches. Superposition is used to compute the contribution from vorticity of separate meshes. The multiresolution algorithm is implemented in two dimensions but is not restricted to two dimensions.

Penalization can be carried out as an interpolation between the solid and fluid velocities in a manner similar to the semi-implicit penalization method. This interpolating penalization eliminates the penalization parameter. The implementation shows good results when applied the impulsively started flow past fixed and rotating circular cylinders. The flow past a suspension bridge including railings and crash barriers demonstrate the ability of proposed multiresolution VIC algorithms to handle complex geometries.

Chapter 4

Three-dimensional particle-mesh method and multiresolution

While two-dimensional simulations are fast and applicable to a variety of flow problems they are insufficient in other cases. E.g. three-dimensional simulations are required for flow around three-dimensional compact objects, bridge decks with strong spanwise variations and flow phenomena that are inherently three-dimensional, e.g. vortex-shedding.

This chapter describes the extension of the vortex-in-cell (VIC) algorithm from two to three dimensions. A periodic and free-space Poisson solver implemented in the parallel particle mesh (PPM) library is presented and validated. The existing `wvic` VIC implementation is periodic. `naga`, a new free-space PPM VIC implementation prepared for multiresolution, is presented and validated. The modifications to PPM required for the current VIC multiresolution algorithm have not been completed, but considerations on multiresolution in PPM are declared.

4.1 Vortex-in-cell method

The vortex-in-cell algorithm requires few modifications going from two to three dimensions. The vorticity is no longer limited to the in-plane component but is a vector

$$\boldsymbol{\omega} = \boldsymbol{\nabla} \times \boldsymbol{u}. \tag{4.1.1}$$

This introduces an additional term, $\boldsymbol{\omega} \cdot \boldsymbol{\nabla} \boldsymbol{u}$, the stretching term to the vorticity transport equation

$$\frac{D\boldsymbol{\omega}}{Dt} = \boldsymbol{\omega} \cdot \boldsymbol{\nabla} \boldsymbol{u} + \nu \boldsymbol{\nabla}^2 \boldsymbol{\omega} \tag{4.1.2}$$

as the inner product is not zero in three dimensions. ν is the kinematic viscosity. The stretching term is discretised in its classical formulation [112]. The flow is defined by its vorticity which is discretised onto particles moving with the fluid velocity $\boldsymbol{u}$. The strength of the particle is defined as the integral of vorticity

$$\boldsymbol{\Gamma} = \int_{V_p} \boldsymbol{\omega} \boldsymbol{d} V \tag{4.1.3}$$

over the particle volume V_p. From Eq. (4.1.2) and $\boldsymbol{u}$ the particle strengths and positions are advanced at discrete time steps with temporal spacing Δt using $\mathcal{O}(2)$ Runge-Kutta time integration. Higher order integration schemes with low storage requirements exist [110] but have not been implemented in the current work. To ensure sufficient support of the interpolation and stability of the diffusion term the size of the time step Δt is limited by

$$\Delta t < C_1 \frac{\Delta x^2}{\nu} \tag{4.1.4}$$

$$\Delta t < C_2 \left|\left| S_{ij} \right|\right|_\infty \tag{4.1.5}$$

respectively. S_{ij} is the flow strain rate and Δx the mesh spacing.

The velocity

$$\boldsymbol{u} = \boldsymbol{\nabla} \times \boldsymbol{\Psi} + \boldsymbol{U}_\infty \tag{4.1.6}$$

is defined as the sum of the curl of the three-dimensional stream function $\boldsymbol{\Psi}$ and an irrotational free-stream velocity $\boldsymbol{U}_\infty$. The stream function is known as the vector potential in e.g. electrodynamics. The volume flow Q through a surface equals the circulation of the stream function

$$Q = \oint_C \boldsymbol{\Psi} \cdot \boldsymbol{ds} \tag{4.1.7}$$

along the corresponding boundary C. Substituting Eq. (4.1.6) into Eq. (4.1.1), and noting that Ψ can be offset to be solenoidal (see Appendix D), yields the Poisson equation

$$-\nabla^2 \boldsymbol{\Psi} + \boldsymbol{\nabla} \left(\boldsymbol{\nabla} \cdot \boldsymbol{\Psi} \right) = \boldsymbol{\omega} \tag{4.1.8}$$

$$-\nabla^2 \boldsymbol{\Psi} = \boldsymbol{\omega} \tag{4.1.9}$$

relating the stream function to the vorticity field. The velocity can subsequently be computed from Eq. (4.1.6).

The Poisson equation is solved using a Green's function solution

$$\boldsymbol{\Psi}(\boldsymbol{x}) = \int_V G(\boldsymbol{y} - \boldsymbol{x})\boldsymbol{\omega}(\boldsymbol{x}) d\boldsymbol{y} = G \circ \boldsymbol{\omega} \tag{4.1.10}$$

where G is the Green's function and $\circ$ signifies convolution over the domain V. G depends on the dimensions and boundary conditions of the computational domain. For a three-dimensional, free-space domain [4]

$$G = \frac{1}{4\pi|\boldsymbol{x}|}. \tag{4.1.11}$$

By *free-space* boundaries the solution is understood to approach zero as $|\boldsymbol{x}| \to \infty$. Interpolating the particle vorticity to an underlying regular mesh using the $M'4$ kernel Eq. (3.1.8) allows fast Fourier transforms (FFT) to be applied to the vorticity. It furthermore eases computation of the RHS of Eq. (4.1.2) and facilitates remeshing the particle from the mesh to ensure regular distribution of particles during the simulation. The convolution can be computed in Fourier space as a product

$$\hat{\boldsymbol{\Psi}} = \hat{G}\hat{\boldsymbol{\omega}}. \tag{4.1.12}$$

Due to the FFT the computational work scales $\mathcal{O}(N \log(N))$, where N is the number of computational elements. The FFT is a discrete Fourier transform and computes a finite numerical integral instead of the infinite integral of the Fourier transform. By contrast, evaluating the convolution integral, Eq. (4.1.10), directly involves N^2 operations. As part of the present work an FFT based routine for solving the Poisson equation has been implemented in PPM. The routine, "`ppm_module_Poisson`", is validated in section 4.2.

Using the FFT implies a periodicity of the input fields and imposes this periodicity to the solution to the spectral Poisson equation, Eq. (4.1.12). This is known as circular convolution. If the domain is periodic the Green's function can be obtained directly from the Fourier transformed Poisson equation

$$-\left(k_x^2 + k_y^2 + k_z^2\right)\hat{\boldsymbol{\Psi}} = \hat{\boldsymbol{\omega}} \tag{4.1.13}$$

where the spatial derivatives are expressed by the corresponding spectral coordinates k_x, k_y, k_z, cf. Appendix B.4. It is seen that the Greens function is given spectrally as

$$\hat{G} = \frac{1}{k_x^2 + k_y^2 + k_z^2}. \tag{4.1.14}$$

Similarly it is possible to compute differential operators such as divergence, the gradient or the curl in Fourier space.

To calculate the free-space solution the vorticity domain is doubled and padded in each spatial direction thereby cancelling the periodicity. To obtain sufficient far-field information of the Green's function it is initialised in this extended domain, as described in section 3.1.

It is possible to have mixed domain boundaries, i.e. one periodic dimension and two unbounded dimensions. This requires solving the one-dimensional Poisson equation in the periodic dimension and using the result as input for a series of unbounded Helmholtz equations [10].

Numerical solution to Eq. (4.1.12) entails three potential error sources. Firstly both Green's functions Eqs. (4.1.14) and (4.1.11) are singular and $G(\mathbf{0})$ must be replaced or approximated by finite values. Replacing the singularity of the spectral Green's function with a finite value corresponds to specifying a mean value of the real space Green's function. This offset is insignificant cf. Eq. (4.1.6). However, $G(\mathbf{0})$ has no such significance and therefore finite convergence must be expected when determining a finite value for $G(\mathbf{0})$. Secondly the smoothness of the function to be transformed affects the spatial convergence of the Fourier modes to the function: The number of terms/frequencies in the FFT Fourier series is proportional to the spatial discretisation and thus the number of truncated terms depends on the resolution. Parsevals identity states that the absolute values of the Fourier coefficients sum to a finite value. Thus the coefficients of the truncated Fourier modes tend to zero for higher frequencies. The smoother the function, the faster high frequency coefficients tend to zero [16]. As the maximum error from the Fourier series to the function is bounded by the sum of the absolute value of the truncated Fourier coefficients [15], discontinuity of the higher order derivatives affects the order of the spatial convergence of the Fourier series to the function. Thirdly the integral of the Fourier transformation is performed numerically via the midpoint rule and may restrict the error of the solution to second order convergence. By careful manipulation [81] the integral of bounded, two-dimensional functions has been evaluated using Simpson's rule resulting in $\mathcal{O}(4)$ error convergence.

The convolution Eq. (4.1.10) corresponds to a filtering and the result, the stream function, can be expected to be smoother than the input. Thus the inverse of transform of $\hat{\mathbf{\Psi}}$ is expected to be associated with a lower error.

4.2 Numerical solution to the Poisson equation

The numerical validation of the solution to the Poisson equation Eq. (4.1.9) is done from the analytic solution

$$\boldsymbol{\Psi}_{\text{target}} = -e^{\sin(xT_x)} \sin(yT_y) \sin(zT_z) \tag{4.2.1}$$

where $T_x = 2\pi/Lx$ is the scaling of a sine wave to the x-extent of the periodic computational domain L_x. Similarly T_y and T_z. Eq. (4.2.1) introduces a continuum of wave numbers in contrast to pure sine-waves, while remaining smooth. The error of the latter is of the order of machine precision no matter the resolution and is therefore not suitable for convergence of the error. Applying the Laplace operator to Eq. (4.2.1) yields

$$\begin{aligned}\boldsymbol{\omega} = e^{\sin(xT_x)} \sin(yT_y) \sin(zT_z) \big[&\sin(xT_x)T_x^2 \\ &- \cos(xT_x)^2 T_x^2 + T_y^2 + T_z^2\big]\end{aligned} \tag{4.2.2}$$

The singularity of the periodic Green's function Eq. (4.1.14) is replaced by $\hat{G}(\mathbf{0}) = 0$ thereby imposing zero mean to both G and $\boldsymbol{\Psi}$. The velocity does not depend on the level of the stream function as the velocity is defined as the curl of the stream function Eq. (4.1.6). Initialising Eq. (4.2.2) in a $1 \times 1 \times 1$ domain on a Cartesian equidistant mesh with N_x points in all three coordinate directions the Poisson equation Eq. (4.1.9) is solved to obtain $\boldsymbol{\Psi}$. As expected the error

$$\epsilon_\Psi = \frac{|\boldsymbol{\Psi}_{\text{comp}} - \boldsymbol{\Psi}_{\text{target}}|_\infty}{|\boldsymbol{\Psi}_{\text{target}}|_\infty} \tag{4.2.3}$$

converges spectrally with respect to N_x as Fig. 4.1a shows. When the numerical error is limited by machine precision it can be improved no further.

The velocity $\boldsymbol{u}$ is computed by applying the curl $\boldsymbol{\nabla}\times$ to the stream function $\boldsymbol{\Psi}$, cf. Eq. (4.1.6). This can be done in Fourier space (see Appendix B.4). The error

$$\epsilon_u = \frac{|\boldsymbol{u}_{\text{comp}} - \boldsymbol{\nabla} \times \boldsymbol{\Psi}_{\text{target}}|_\infty}{|\boldsymbol{\nabla} \times \boldsymbol{\Psi}_{\text{target}}|_\infty} \tag{4.2.4}$$

between the numerical solution and the curl of Eq. (4.2.1) converges spectrally in contrast to a $\mathcal{O}(4)$ finite difference (FD) approximation as Fig. 4.1b shows.

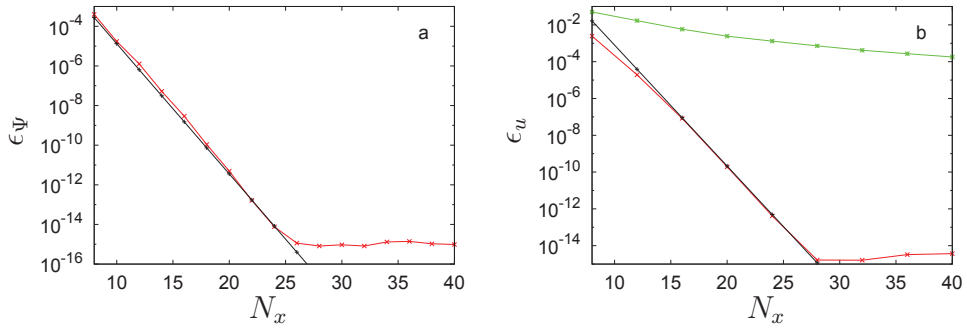


Figure 4.1: (a) The error (red) of the solution to the Poisson equation Eq. (4.2.1) in a periodic domain as a function of N_x. 0.22^{N_x} is shown as a reference (black). (b) The error of the corresponding velocity with curl determined spectrally (red) and curl from $\mathcal{O}(4)$ finite differences (green).

4.2.1 Numerical solution to the Poisson equation in an unbounded domain

The real valued Green's function is singular and $G(\mathbf{0})$ must be replaced by a finite value to permit numerical solution. In this work the singularity has mainly been treated in real space. Substituting $G(\mathbf{0})$ by a large numerical value generates large errors and is not feasible. $G(\mathbf{0})$ should be chosen to accommodate the discrete Fourier transformation and the finite difference operators. Different approaches are examined in the following. Furthermore the discrete G will, due to the periodicity of the FFT, experience a discontinuity of the first derivative at the periphery of the initialised G. This will result in finite convergence. It is conjectured that the first derivative may be made continuous by expanding the $\boldsymbol{\omega}$ and G fields beyond the zero-padding into a connection region. In the connection region $\boldsymbol{\omega}$ is set to zero and G is modified to be continuous to a specific order. Similar to the free-space zero-padding the connection region does not contribute to the integral of Eq. (4.1.10). This has not been attempted as part of the present work.

To test the numerical implementation and evaluate the influence of the value of $G(\mathbf{0})$ a suitable benchmark is required. During the present work no benchmark for the spatial convergence of the free-space solution literature has been encountered. Constructing a suitable benchmark in a three-dimensional free-space domain that does not requires time integration of associated physics requires some consideration. To ensure that the vorticity field is solenoidal in the three-dimensional unbounded domain vortex filaments must form closed loops cf. Helmholtz' second theorem. The simplest

form fulfilling this requirement is the vortex ring. It follows from Eq. (4.1.9) that the vorticity is solenoidal and compact if the stream-function is.

Therefore a compact, bounded and ring shaped stream-function is sought as the corresponding vorticity field is easily derived. Gaussian or sinusoidal cores are apparent choices. However, specifying a Gaussian decay of the stream function entails a singularity in origin of the corresponding vorticity field. For a ring shaped stream function decaying sinusoidally to zero the corresponding vorticity is discontinuous due to the radial term of the $\boldsymbol{\nabla}$ operator in cylindrical coordinates. Instead the bump function

$$f_B(r) = \begin{cases} e^{-\frac{c_B}{1.0-r^2}} & \text{if } |r| < 1 \\ 0 & \text{otherwise} \end{cases}, \tag{4.2.5}$$

where $c_B = 10$ is a constant, is proposed. It is finite valued, has compact support with a core radius of 1 and is infinitely differentiable. These properties apply to the stream function by using f_B as the core of a torus shaped stream function. The azimuthal component of the stream function is the only non-zero component

$$\Psi_{\text{target},\phi}(\rho, \phi, z) = f_B\left(\sqrt{(\rho - R)^2 + z^2}\right), \tag{4.2.6}$$

where $R = 1$ is the radius of the ring. The corresponding vorticity field is given by applying Eq. (4.1.9) to $\boldsymbol{\Psi}$

$$\boldsymbol{\omega}(\rho, \phi, z)_{\text{target}} = -\nabla^2 \boldsymbol{\Psi}_{\text{target}}. \tag{4.2.7}$$

The corresponding velocity field is determined from Eq. (4.1.6)

$$\boldsymbol{u}_{\text{target}}(\rho, \phi, z) = \nabla \times \boldsymbol{\Psi}_{\text{target}}. \tag{4.2.8}$$

Both Eqs. (4.2.7) and (4.2.8) exist in closed and can be found in Appendix E. Fig. 4.2 shows the magnitude of the stream function and vorticity. For the lowest resolutions in the following analysis the large gradients obviously will be improperly resolved.

For the numerical validation a $4 \times 4 \times 4$ domain is uniformly resolved by $N_x \times N_x \times N_x$ mesh points. The vorticity field is initialised and as reference the solution to the Poisson equation and the corresponding velocity field is initialised. The velocity $\boldsymbol{u}_{\text{comp}}$ is computed by applying the $\mathcal{O}(2)$ or $\mathcal{O}(4)$ FD curls to $\boldsymbol{\Psi}_{\text{comp}}$ and the errors ϵ_Ψ and ϵ_u are evaluated cf. Eqs. (4.2.3) and (4.2.4). In one configuration a significant reduction of the error level is obtained from spectral computation of the curl. Otherwise spectral computation of the curl does not improve the error level or convergence.

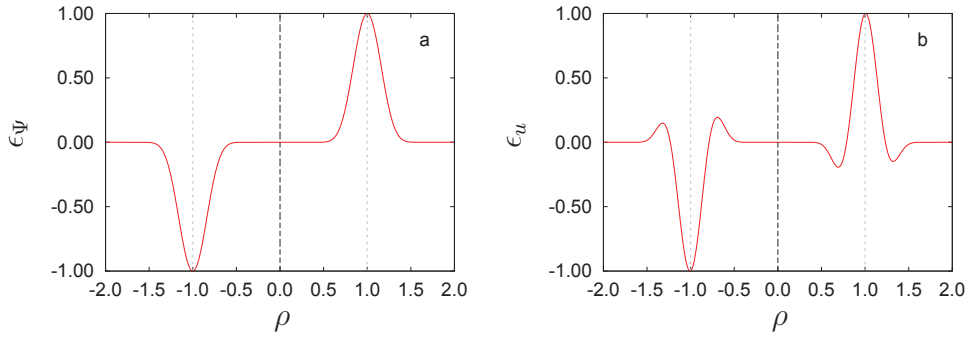


Figure 4.2: Magnitude of (a) $\mathbf{\Psi}_{\text{target}}$ and (b) the corresponding vorticity Eq. (4.2.7) normalised by their respective maximum value. For $\theta = 0$, $\phi = 0$.

4.2.1.1 Hockney & Eastwood, Chatelain & Koumoutsakos

Hockney and Eastwood [46] propose substituting the singularity of the real valued Green's function by

$$G(\mathbf{0}) = 1. \tag{4.2.9}$$

This adds the value 1 to all wave amplitudes in Fourier space similar to the analytic solution to the Fourier transform of the Dirac delta. Chatelain and Koumoutsakos [10] propose setting

$$G(\mathbf{0}) = \left(\frac{3\Delta x\Delta y\Delta z}{4\pi}\right)^{2/3}. \tag{4.2.10}$$

to an approximation to the integral of G over the cell containing the singularity, where Δx,Δy,Δz are the mesh spacing in the x,y,z directions. Neither Eq. (4.2.9) nor Eq. (4.2.10) scale $\propto 1/\Delta x$ and therefore jumps can be expected as the resolution increases since the discrete values of G in the vicinity of the singularity are determined by Eq. (4.1.11). Fig. 4.3a shows the error ϵ_Ψ as a function of N_x. For both values of $G(\mathbf{0})$ the error converges second order. The order of convergence is not well documented in the literature but $\mathcal{O}(2)$ is commonly expected for the free-space VIC. Fig. 4.3b shows the error of the velocity computations as a function of N_x. As expected the error convergence remains second order for both values of $G(\mathbf{0})$ and both FD stencils, though the error of the $\mathcal{O}(4)$ stencil is consistently lower than that of the $\mathcal{O}(2)$ stencil.

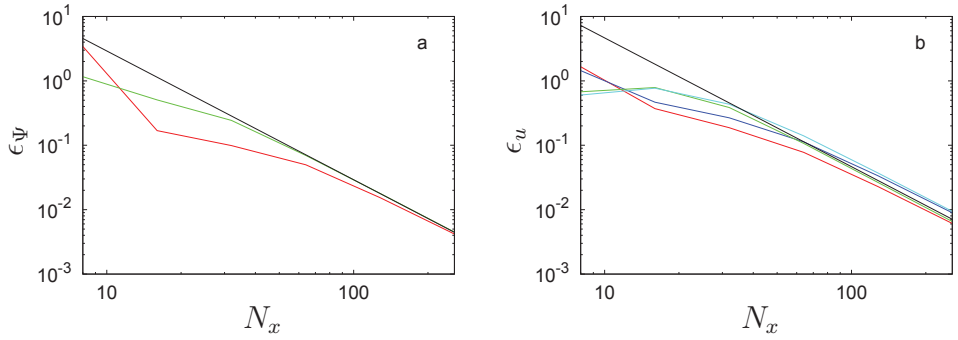


Figure 4.3: (a) Error of the solution to the Poisson equation and (b) velocity computation (using $\mathcal{O}(4)$ FD) treating the singularity as $G(\mathbf{0}) = 1$ (red) or cf. Eq. (4.2.10) (green), as a function of N_x. Errors of velocity computation using $\mathcal{O}(2)$ FD curl, (blue)/(cyan) respectively. N_x^{-2} is shown as a reference (black).

4.2.1.2 Finite difference based estimates

As finite difference approximations to e.g. derivatives are based on weighted grid values any estimate of $G(\mathbf{0})$ based on finite differences will be remain proportional to the adjacent grid points, thereby preventing discontinuity around the discrete point containing $G(\mathbf{0})$.

By extrapolating the values of the Green's function of the point adjacent to the singular $G(\mathbf{0})$ point and estimate for $G(\mathbf{0})$ is obtained. For this a finite difference approximation is used, see Appendix C.1. E.g. for a $\mathcal{O}(6)$ extrapolation

$$\begin{aligned} G(\mathbf{0}) &= \frac{6}{4\pi\Delta x} - \frac{15}{4\pi 2\Delta x} + \frac{20}{4\pi 3\Delta x} - \frac{15}{4\pi 4\Delta x} + \frac{6}{4\pi 5\Delta x} - \frac{1}{4\pi 6\Delta x} \\ &= \frac{49}{80\pi\Delta x}. \end{aligned} \tag{4.2.11}$$

This value scales with $1/\Delta x$ and therefore does not introduce jumps in G at any resolution.

Another finite difference based approach is to determine $G(\mathbf{0})$ such that the correspondence of the finite difference based curl of the stream function to the expected velocity field is improved. Given a unit strength point vortex, the analytic solution to the derivative of the stream function is known; it equals G. In each point where $G(\mathbf{0})$ is used in the FD stencil an estimate for $G(\mathbf{0})$ can be made. Only in points aligned along the coordinate axes is the finite difference evaluation based on the value $G(\mathbf{0})$. And only within the

$G(\mathbf{0})_{\text{FD order,point}}$	$G_{2,1}$	$G_{4,1}$	$G_{4,2}$	$G_{6,1}$	$G_{6,2}$	$G_{6,3}$
Exact value	$\frac{5}{8\pi\Delta x}$	$\frac{25}{48\pi\Delta x}$	$\frac{31}{48\pi\Delta x}$	$\frac{39}{80\pi\Delta x}$	$\frac{41}{80\pi\Delta x}$	$\frac{48}{80\pi\Delta x}$
$G(\mathbf{0})\Delta x$ (rounded)	0.1989	0.1658	0.2056	0.1552	0.1631	0.1950

Table 4.1: Values of $G(\mathbf{0})$ fitted to FD difference stencils.

extent of the FD stencil. E.g. for an $\mathcal{O}(2)$ FD stencil around the ith point

$$\left.\frac{\partial G}{\partial x}\right|_i = \frac{G[i+1] - G[i-1]}{2\Delta x}. \tag{4.2.12}$$

If $i = 1$ and $G(\mathbf{0})$ is in point 0 then

$$\left.\frac{\partial G}{\partial x}\right|_1 = \frac{G(2\Delta x) - G(\mathbf{0})}{2\Delta x} = -\frac{1}{4\pi\Delta x^2} \tag{4.2.13}$$

which is solved for

$$G(\mathbf{0}) = G_{2,1} = \frac{5}{8\pi\Delta x}. \tag{4.2.14}$$

The subscript denotes that the value is based on the $\mathcal{O}(2)$ finite difference stencil in the first point off the singularity. For the $\mathcal{O}(4)$ FD stencil two values for $G(\mathbf{0})$ are obtained and for $\mathcal{O}(6)$ FD stencil three values are obtained, see Tab. 4.1. For $G_{4,1}$, $G_{6,1}$ and $G_{6,2}$ the calculated $G(\mathbf{0})$ will include compensation for the points on the opposite side of the singularity on which the FD approximation is also based. The estimates, $G_{2,1}$, $G_{4,2}$ and $G_{6,3}$, depend on $G(\mathbf{0})$ as the outermost point of the FD stencil. These values of $G(\mathbf{0})$ deviate by less than 6%, cf. Table 4.1.

Each estimate for $G(\mathbf{0})$ is optimised for a particular point relative to the singularity. The error in the neighbouring points can easily be evaluated for all estimates of $G(\mathbf{0})$. Based on this an optimum weight between e.g. $G_{4,1}$ and $G_{4,2}$ can be determined:

$$G_{4,\text{opt}} = \frac{8/9}{G_{4,1}} + \frac{1/9}{G_{4,2}}. \tag{4.2.15}$$

However, these values have been found not to be optimal. As is the case for the $\mathcal{O}(6)$ extrapolation to $G(\mathbf{0})$ (Eq. (4.2.11)), $G(\mathbf{0})$ fitted to the FD derivatives scales $\propto 1/\Delta x$ with respect to the resolution. That is, these values differ only by a factor.

Fig. 4.4a shows the error of the solution to the Poisson equation. Using $G(\mathbf{0}) = G_{4,1}, G_{4,2}$ and Eq. (4.2.11) the error converges $\mathcal{O}(2)$. However the

level of the error is lowered as much as one order of magnitude compared to solutions with $G(\mathbf{0}) = 1$. The error of the velocity computations converges similarly to that of the Poisson equation, see Fig. 4.4b. Even though $G(\mathbf{0}) = G_{4,1}$ and $G(\mathbf{0}) = G_{4,2}$ have been formulated from considerations of the derivative of the stream function there is no additional improvement of the velocity error.

In search of an expected optimal error level with $\mathcal{O}(2)$ convergence the weight $G_{4,\text{opt}}$ has been used with no improvement in neither error level nor convergence rate (not shown). Several $G(\mathbf{0})$ values in the $G_{2,1}$ to $G_{6,3}$ vicinity have been used resulting in insignificant variation of the level of the error. By shear coincidence

$$G(\mathbf{0}) = G_C = \frac{34}{48\pi\Delta x} = G_{4,2} + \frac{G_{4,2} - G_{4,1}}{2} \tag{4.2.16}$$

has been considered instead of the mean of $G_{4,1}$ and $G_{4,2}$

$$\frac{G_{4,1} + G_{4,2}}{2} = G_{4,2} - \frac{G_{4,2} - G_{4,1}}{2}. \tag{4.2.17}$$

As seen on Fig. 4.4, using G_C yields roughly $\mathcal{O}(4)$ error convergence for both the solution to the Poisson equation and the velocity computation. The similarly improved solution to the Poisson equation underlines the coincidence of the value of G_C. The result is a significantly lowered error at the higher resolutions. By varying G_C by a factor of just 5% returns $\mathcal{O}(2)$ convergence of the error. This inexplicable improved convergence of the error may be a local optimum. Assuming that the $1/\Delta x$ scaling generally is beneficial any global optimum is not easily found due the very high sensitivity of the prefactor. Using G_C while applying the curl spectrally reduces the error level almost an additional order of magnitude, Fig. 4.4b. For all other Green's functions computing the curl spectrally offers at best a slight improvement of the error at low resolution.

For $G(\mathbf{0}) = G_C$ the error of the solution to the Poisson equation and the velocity computation deviates from the $\mathcal{O}(4)$ trend line around $N_x = 256$. This could be due to any of the FFT related error sources in section 4.1 including the finite valued singularity. The increased deviation of the error of the stream function may be due to an incorrect level of the stream function which does not affect the velocity solution.

4.2.1.3 Regularised Green's function

For direct particle-particle interaction the singularity of the Green's function may be resolved but requiring very small time steps for the time integration.

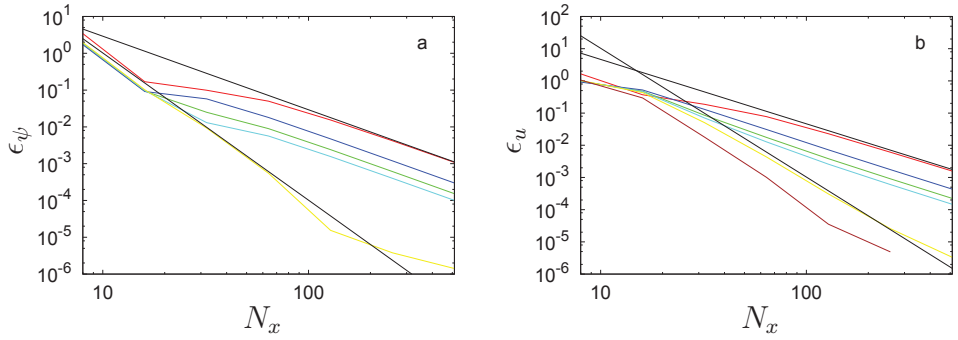


Figure 4.4: (a) Error of the solution to the Poisson equation or (b) velocity computation (using $\mathcal{O}(4)$ FD curl) with various $G(\mathbf{0})$ values: Eq. (4.2.11) (green), $G(\mathbf{0}) = G_{4,1}$ (blue), $G(\mathbf{0}) = G_{4,2}$ (cyan) and Eq. (4.2.16) using $\mathcal{O}(4)$ FD (yellow) and spectral curl (brown). Errors using $G(\mathbf{0}) = 1$ (red) and N_x^{-2}/N_x^{-4} (both black) are shown as a references.

Instead the vorticity of the particle can be made finite valued by convolution with the radially symmetric core $\zeta(\rho)$ which must

$$4\pi \int_0^\infty \zeta(\rho)\rho^2 d\rho = 1. \tag{4.2.18}$$

The spatial distribution of the core is scaled by a radius $\sigma \propto \Delta x$, where Δx is the mesh spacing. To ensure that the core integrates to 1, $\zeta_\sigma(\boldsymbol{x}) = \zeta(|\boldsymbol{x}|/\sigma)/\sigma^3$. Due to the associativity and commutativity of the convolution the regularisation may be applied to the Green's function which in turn will be convolved with the vorticity to obtain the stream function:

$$\boldsymbol{\Psi} = G \circ (\zeta_\sigma \circ \boldsymbol{\omega}) = (\zeta_\sigma \circ G) \circ \boldsymbol{\omega} \tag{4.2.19}$$

$$\boldsymbol{\Psi} = G_\sigma \circ \boldsymbol{\omega}. \tag{4.2.20}$$

If $\sigma \propto \Delta x$ the error of the regularised velocity computation converges $\mathcal{O}(\Delta x^m)$ [5], where m is related to the number of even moments of ζ_σ.

The error of the regularised Green's function [112]

$$G_\sigma^2 = \frac{\operatorname{erf}\left(\frac{|\boldsymbol{x}|/\sigma}{\sqrt{2}}\right)}{4\pi|\boldsymbol{x}|} \tag{4.2.21}$$

scales $\mathcal{O}(2)$ and the error of

$$G_\sigma^4 = \frac{\operatorname{erf}\left(\frac{|\boldsymbol{x}|/\sigma}{\sqrt{2}}\right) + \frac{1}{\sqrt{2\pi}}|\boldsymbol{x}|e^{-(|\boldsymbol{x}|/\sigma)^2/2}}{4\pi|\boldsymbol{x}|} \tag{4.2.22}$$

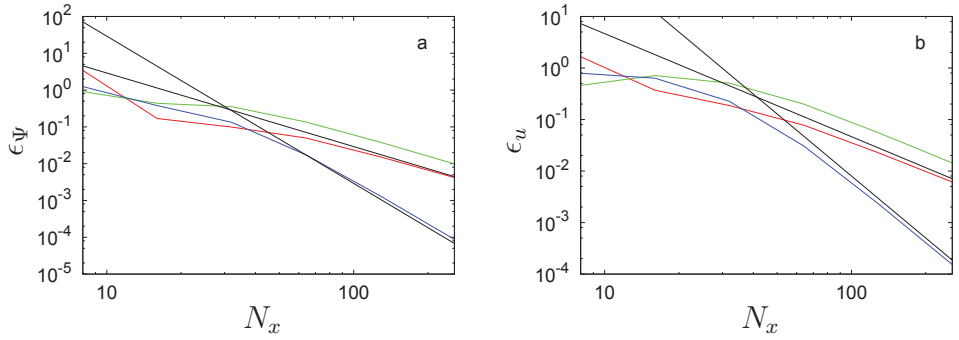


Figure 4.5: (a) Error of the solution to the Poisson equation or (b) velocity computation (using $\mathcal{O}(4)$ FD curl) using the $\mathcal{O}(2)$ and $\mathcal{O}(4)$ regularised Green's functions, (green) and (blue) respectively. Errors using $G(\mathbf{0}) = 1$ (red) and N_x^{-2} and N_x^{-4} (black) are shown for comparison.

scales $\mathcal{O}(4)$.

For the regularised Green's function, G_σ^2, Fig. 4.5a shows the error of the solution to the Poisson equation. The optimal scaling of the error is obtained if $\sigma/\Delta x = 1$ [5]. As expected the error converges $\mathcal{O}(2)$ but at a higher level than for $G(\mathbf{0}) = 1$; at low resolution the regularised Green's function is heavily modified compared to Eq. (4.1.11) and as the convergence order is expected to be $\mathcal{O}(2)$ the level of the error remains higher. Using G_σ^4 lowers the error and improves the rate of convergence to the $\mathcal{O}(4)$ of its design.

The error of the velocity computed by $\mathcal{O}(4)$ FD stencils converges similar to the error of the solution to the Poisson equation, see Fig. 4.5b.

4.2.1.4 Regularised velocity kernel

The velocity may be calculated directly by applying the curl of the stream function analytically

$$\boldsymbol{u} = \boldsymbol{\nabla} \times (G \circ \boldsymbol{\omega}) = \boldsymbol{K} \circ \boldsymbol{\omega} \tag{4.2.23}$$

where

$$\boldsymbol{K} = -\frac{1}{4\pi|\boldsymbol{x}|^3}\boldsymbol{x}\times \tag{4.2.24}$$

is the velocity kernel. This kernel is, like the Green's function, singular but may similarly be regularised [5, 112] (cf. Appendix F). The kernels of Table F.1 $\boldsymbol{K}_\sigma^{(4)}, \boldsymbol{K}_\sigma^{(6)}, \boldsymbol{K}_\sigma^{(8)}, \boldsymbol{K}_\sigma^{(10)}$ respectively of order $\mathcal{O}(4), \mathcal{O}(6), \mathcal{O}(8), \mathcal{O}(10)$ have been derived. Fig. 4.6 shows the error of the velocity computation using

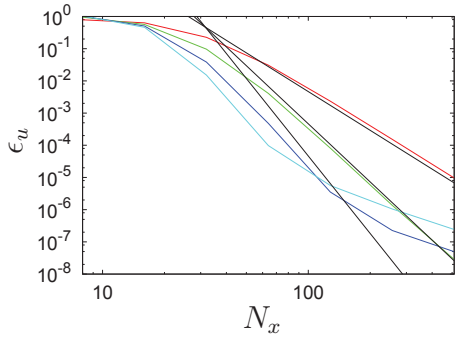


Figure 4.6: Error of the velocity computation using the regularised velocity kernels $\boldsymbol{K}_\sigma^{(4)}$ (red), $\boldsymbol{K}_\sigma^{(6)}$ (green), $\boldsymbol{K}_\sigma^{(8)}$ (blue) and $\boldsymbol{K}_\sigma^{(10)}$ (cyan) and N_x^{-4},N_x^{-6}, N_x^{-8} (black) are as shown reference.

the velocity kernels. $\boldsymbol{K}_\sigma^{(4)}$ yields $\mathcal{O}(4)$ convergence and a lower error than all Green's functions except Eq. (4.2.16). $\boldsymbol{K}_\sigma^{(6)}$ converges $\mathcal{O}(6)$ and at $N_x = 512$ provides the lowest error to the velocity computation. Computing the velocity using the $\boldsymbol{K}_\sigma^{(8)}$ or $\boldsymbol{K}_\sigma^{(10)}$ kernels yields very low errors for $N_x \leq 256$ but beyond this resolution $\boldsymbol{K}_\sigma^{(6)}$ is preferable. $\boldsymbol{K}_\sigma^{(8)}$ or $\boldsymbol{K}_\sigma^{(10)}$ do not achieve the $\mathcal{O}(8)$ and $\mathcal{O}(10)$ convergence of their respective design, but instead seem to be converge $\mathcal{O}(2)$ at high resolution. The gradients of the kernels increase for increasing order of the kernel. Thus the kernel is under-resolved on the mesh if the regularisation function is scaled with $\sigma/\Delta x = 1$. This has not been investigated further in the present work.

Obviously $\boldsymbol{K}$ only solves for the velocity field and the impressive convergence applies only the velocity computation on the grid. Though other factors, such as interpolation, may be limiting spatial convergence at high resolution the velocity is solved for with a lower error. If a solution to the pure Poisson equation is required the Green's function must also be initialised.

4.2.2 Free-space using the periodic Green's function

The error level and convergence rate is strongly dominated by the treatment of the singularity. When replacing the singularity of the periodic Green's function a value can be chosen freely as it modifies the level of the solution which in turn is eliminated by the curl Eq.4.1.6. It is conjectured that spectral initialisation of the free-space Green's function is required for spectral error convergence.

The periodic Green's function can be applied to free-space domains by

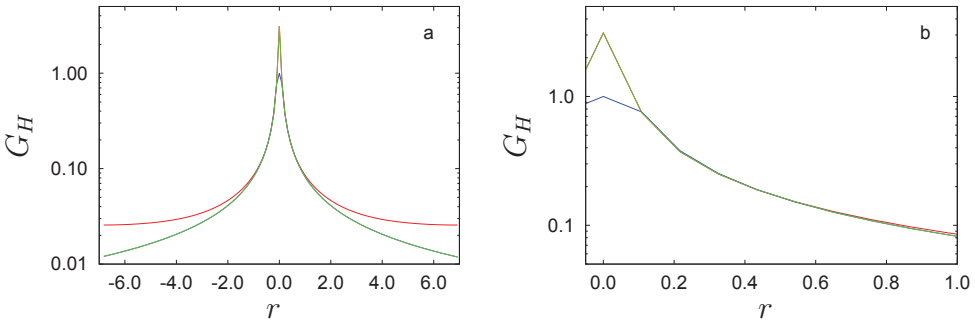


Figure 4.7: The periodic Green's function (red), the free-space Green's function (blue) mostly covered by the hybrid Green's function (green) as a function of radial distance r along the diagonal of the cubic domain. (b) A close-up of the region around $G(\mathbf{0})$. G_H follows the periodic Green's function until $R_H = 10\Delta x) = 0.625$.

arguing that the close-range interaction between vortices is stronger than the far-field. Thus the increased memory used for the free-space solution may used differently; the extension of the domain may be applied in a fully periodic simulation separating the images from the area of interest significantly. If one can make this compromise spectral convergence - of the periodic solution - is obtained at roughly the same hardware requirements.

Following the discussion above the close-range shape of the periodic Green's function is expected to be similar to that of the free-space Green's function. The prospect is that the spectrally initialised periodic Green's function is compatible with the FFT and thus give an optimal value for $G(\mathbf{0})$ and the adjacent points. Therefore the core of the Green's function Eq. (4.1.11) is replaced by the periodic Green's function, until a radial cut-off $R_H(\Delta x)$. The hybrid Green's function is denoted G_H. As Fig. 4.7 shows, the periodic and the free-space Green's functions deviate little in the vicinity of $G(\mathbf{0})$. The free-space singularity has been replaced by $G(\mathbf{0}) = 1$ cf. [46]. Any offset between the free-space and periodic Green's function at $R(\Delta x)$ along the coordinate axes is subtracted to ensure a continuous G_H. However, while the free-space Green's function is spherically symmetric this is not the case for the periodic Green's function and G_H is expected to jump at large R_H. As the resolution is increased and the distance $R_H(\Delta x)$ decreases G_H smoothens. At $\Delta x = 4/128$ resolution G_H remains visibly smooth at $R_H = 10\Delta x$ as seen on Fig. 4.7.

Using G_H the error of the solution to the Poisson equation and the error of the velocity solution remains $\mathcal{O}(2)$ convergent, see Fig. 4.8, but at a lower

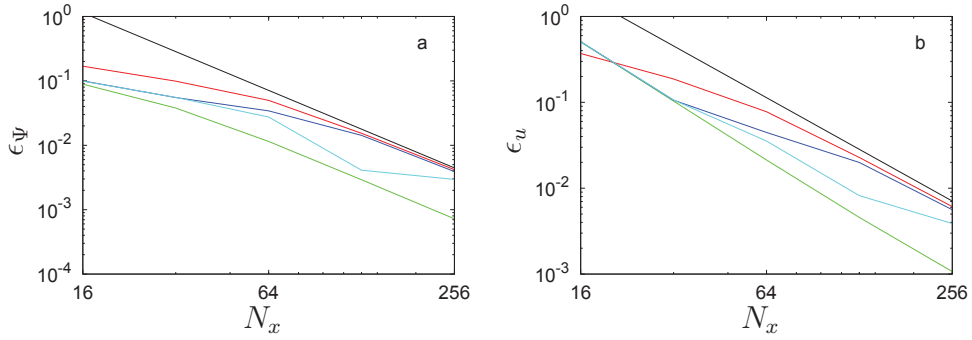


Figure 4.8: (a) Error of the solution to the Poisson equation or (b) velocity computation (using $\mathcal{O}(4)$ FD curl) for $G(\mathbf{0}) = G_H$ with $R_H = \Delta x$ (green), $R_H = 5\Delta x$ (blue) and $R_H = 10\Delta x$ (cyan). Results using $G(\mathbf{0}) = 1$ (red) and N_x^{-2} (black) are shown as references.

level than $G(\mathbf{0}) = 1$ provides. It is highly dependent on R_H and $R_H = \Delta x$ is optimal, corresponding to only replacing the value of $G(\mathbf{0})$.

4.2.3 Divergence of vorticity

Vorticity is by its definition Eq. (4.1.1) solenoidal. For two-dimensional analysis the vorticity field consists of one non-zero component which is constant along that coordinate-direction and is therefore always solenoidal. In three-dimensional analysis this is not automatically ensured as vorticity is the governing variable and not recalculated from the velocity. The rate of change of the vorticity given by the vorticity equation Eq. (4.1.1) is not divergent. However, numerical errors, from e.g. time integration, interpolation, finite difference approximations, etc., may introduce a divergent element to the vorticity.

The contribution of the penalization term in the vorticity equation will be divergence-free up to the precision of the numerical evaluation of the curl operator. This applies whether the penalization is imposed explicitly as a right hand side term or in a split-step fashion like the interpolating penalization. In the velocity-pressure formulation this is not guaranteed if a mollification interval is used.

The divergent part of the vorticity field can be reduced by *vorticity reprojection* [23]. The Helmholtz decomposition

$$\boldsymbol{\omega} = \boldsymbol{\nabla}\phi + \boldsymbol{\nabla} \times \boldsymbol{\psi} \tag{4.2.25}$$

resolves the vorticity by an irrotational and a solenoidal component. Taking

the divergence of Eq. (4.2.25) yields the Poisson equation

$$\nabla^2\phi = \boldsymbol{\nabla}\cdot\boldsymbol{\omega} \tag{4.2.26}$$

which can be solved by convolution by a proper Green's function. Subsequently $\boldsymbol{\nabla}\phi$ can be subtracted from $\boldsymbol{\omega}$. Evaluation of the divergence, convolution with the Green's function and computation and subtraction of the gradient of ϕ is done spectrally.

4.2.3.1 Validation of reprojection implementation

A divergence-free reference vorticity field Eq. (4.2.29) is initialised in a $2 \times 2 \times 2$ periodic domain at a $N_x \times N_x \times N_x$ resolution, see Fig. 4.9a. The vorticity field is overlaid by the gradient of the infinitely differentiable scalar bump function, see Fig. 4.9b,

$$\phi = \begin{cases} -e^{-\frac{10}{1-r^2}} & \text{for } 0 \leq \ r < 1 \\ 0 & \text{for } 1 \leq \ r \end{cases} \tag{4.2.27}$$

where r is the radial distance to the centre of the spherically symmetric function. Fig. 4.10 shows the magnitude of the gradient of Eq. (4.2.27) as a function of r. This addition is highly divergent. The field must be infinitely differentiable otherwise errors of finite convergence are introduced.

By applying vorticity reprojection, the divergent addition is visibly removed, see Fig. 4.9c. Subtracting the reference field (Fig. 4.9a) from the reprojected field (Fig. 4.9c) reveals the error ϵ_ϕ of the reprojection (Fig. 4.9d). The maximum value of ϵ_ϕ converges spectrally until machine precision in the resolution range $N_x \in [16 : 100]$, see Fig. 4.11a.

In a free-space domain a smooth vorticity field, Eq. (4.2.7), is initialised and overlaid by the gradient field of Eq. (4.2.27). Following the procedure above the maximum error of the reprojection is found. Fig. 4.11b shows the reprojection error ϵ_ϕ as a function of N_x using Eq. (4.2.9), G_C and G_σ^4 for the Green's function singularity. The characteristics of the error convergence are similar to those of the free-space solution to the Poisson equation described in section 4.2.1.

4.2.4 Taylor-Green

The periodic Taylor-Green vortices [11] described by the velocity field

$$\boldsymbol{u} = \begin{bmatrix} v_0 \cos(x)\sin(y)\cos(z) \\ -v_0 \sin(x)\cos(y)\cos(z) \\ 0 \end{bmatrix} \tag{4.2.28}$$

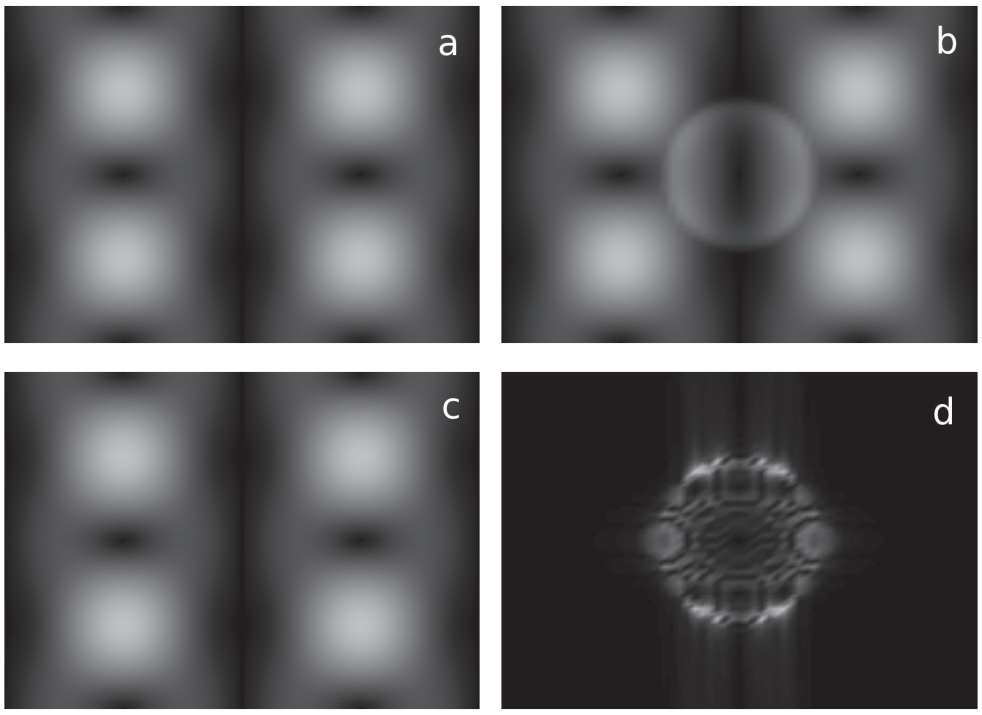


Figure 4.9: Slice through the centre of the domain along the $[1, 1, 0]$ vector, minimum to maximum vorticity magnitude coloured from black to white at $[64 \times 64 \times 64]$ resolution. (a) Magnitude of vorticity for Taylor-Green vortices, (b) magnitude of vorticity after adding $\boldsymbol{\nabla}\phi$, (c) magnitude of vorticity after reprojection, (d) error after vorticity reprojection.

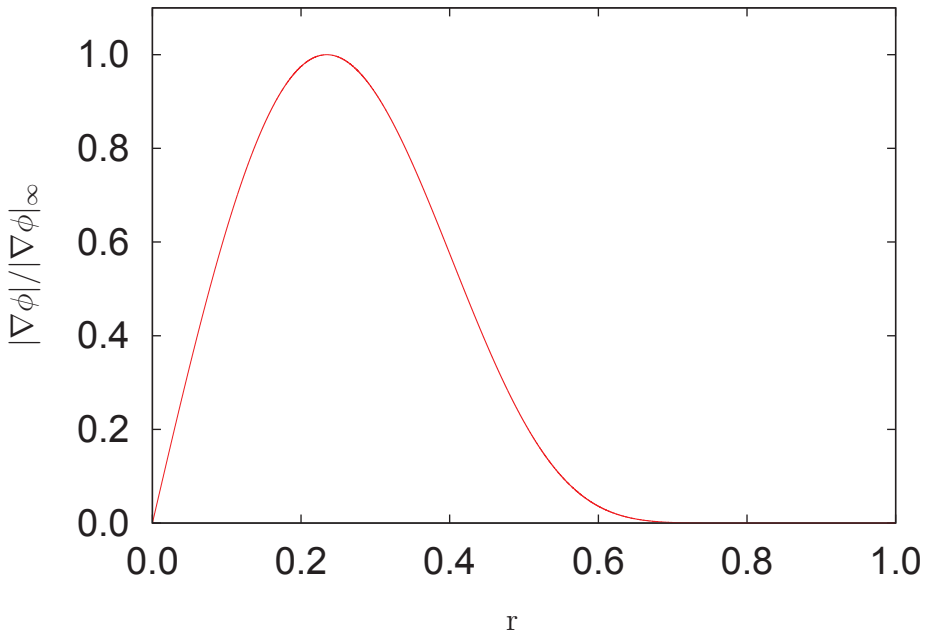


Figure 4.10: Magnitude of the divergent contribution $\boldsymbol{\nabla}\phi$ as function of radial distance r.

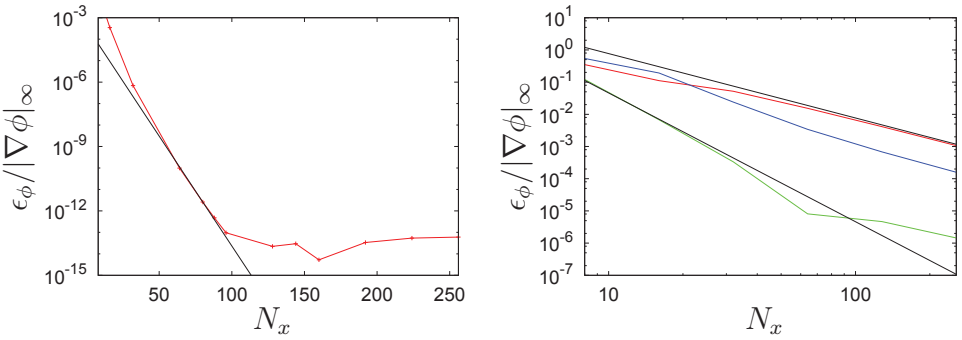


Figure 4.11: (a) Maximum remainder ϵ_ϕ of the divergent contribution $\boldsymbol{\nabla}\phi$ after reprojection as a number of cells N_x along the side of the domain. (a) In a periodic domain with 0.79^{N_x} (black) as a reference. (b) ϵ_ϕ from reprojection in a free-space domain using Eq. (4.2.9) (red), G_C (green) and G_σ^4 (blue) for the Green's function singularity. N_x^{-2} and N_x^{-4} are shown as references (black)

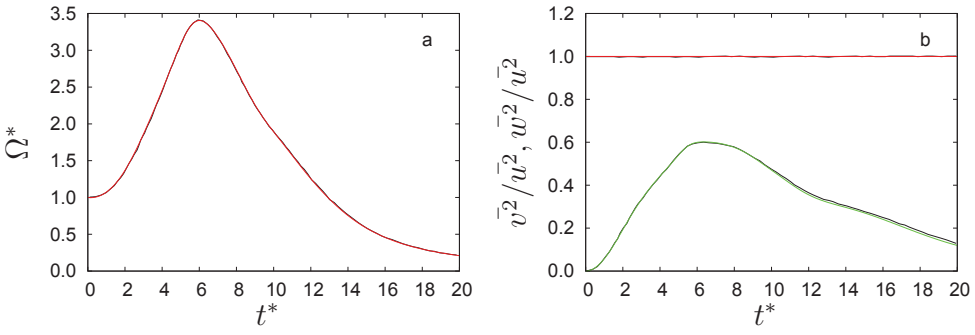


Figure 4.12: Flow simulation of Taylor-Green vortices Eq. (4.2.29) at Re=200. (a) Evolution of normalised enstrophy $\Omega^* = \Omega/\Omega(t=0)$ from the present work (red) and the reference [11] (black). (b) Relative evolution of components of the spatially averaged kinetic energy $\bar{v^2}/\bar{u^2}$ (red), $\bar{w^2}/\bar{u^2}$ (green).

and the corresponding vorticity field

$$\boldsymbol{\omega} = \begin{bmatrix} -v_0 \sin(x)\cos(y)\sin(z) \\ -v_0 \cos(x)\sin(y)\sin(z) \\ -v_0 \cos(x)\cos(y)\cos(z) \end{bmatrix} \tag{4.2.29}$$

is initialised for $v_0 = 1$ in a cubic domain with side length $L = 2\pi$, resolved by N_x points in all directions. The initial flow is well-defined and smooth but as the flow develops the vortex lines are rotated and intensified by vortex stretching, leading to an increase of the enstrophy

$$\Omega = \int_V \boldsymbol{\omega} \cdot \boldsymbol{\omega} dV. \tag{4.2.30}$$

Fig. 4.12a shows the time evolution of enstrophy for $N_x = 128$ and a Reynolds number

$$\mathrm{Re} = \frac{v_0 L}{2\pi\nu} = 200. \tag{4.2.31}$$

Time is non-dimensionalised as

$$t^* = \frac{t 2\pi}{L v_0}. \tag{4.2.32}$$

Vorticity is reprojected every 10 time steps and the size of time step is $\Delta t^* = 0.1$. The correspondence between the present simulations and the spectral method reference [11] is excellent. Fig. 4.12b shows the relative

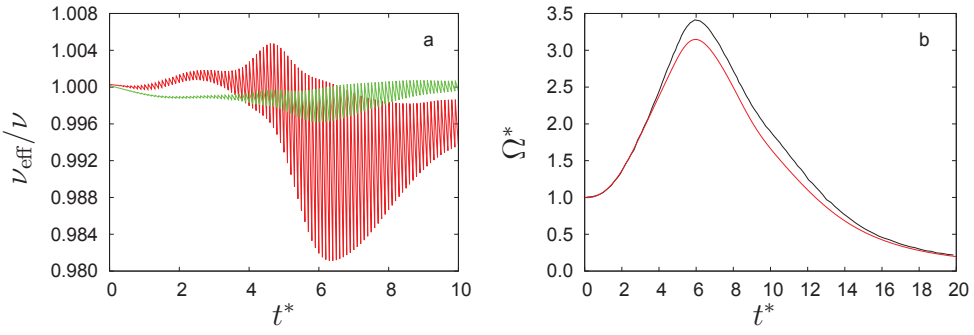


Figure 4.13: Flow simulation of Taylor-Green vortices at Re=200. (a) Normalised effective viscosity ν_{eff}/ν in $128 \times 128\times$ (red) and $256 \times 256 \times 256$ domains. (b) Enstrophy in a simulation where vorticity is replaced by $\boldsymbol{\nabla} \times \boldsymbol{u}$ every time step (red) and the reference [11] (black).

evolution of the components of the spatially averaged kinetic energy $\bar{\boldsymbol{u}}^2$. Again the agreement with [11] is good.

Due to numerical errors the solution may be diffused more than equivalent to the viscosity. The time rate change of kinetic energy and enstrophy give an estimate of the mean effective viscosity [112]

$$\nu_{\text{eff}} = \frac{1}{\Omega}\frac{dE_{\text{kin}}}{dt}. \tag{4.2.33}$$

Fig. 4.13a shows the effective viscosity for $N_x = 128$ and $N_x = 256$. The reprojection interval is clearly visible within the envelope of the effective viscosity. The deviation to the molecular viscosity is less than 2% at the low resolution and is reduced beyond 0.5% at the high resolution. By performing reprojection every time step the deviation is reduced to 0.5% and 0.1% respectively and the time history of the effective viscosity becomes smooth curves (not shown).

To investigate the impact of numerical diffusion an experiment is performed. The vorticity is replaced by the curl of the velocity field computed with $\mathcal{O}(4)$ finite differences. This lowers the integrated magnitude of vorticity (not shown) and the enstrophy considerably, see Fig. 4.13b.

For Re=300 at the same numerical setup the evolution of the enstrophy does not agree with the reference to the same extent, see Fig. 4.14a. If no reprojection is done the enstrophy is considerably higher than the reference, indicating that the divergent part contributes to an increase of the magnitude of the vorticity field. Indeed, without reprojection the integrated magnitude of vorticity increases and the effective viscosity is reduced by roughly

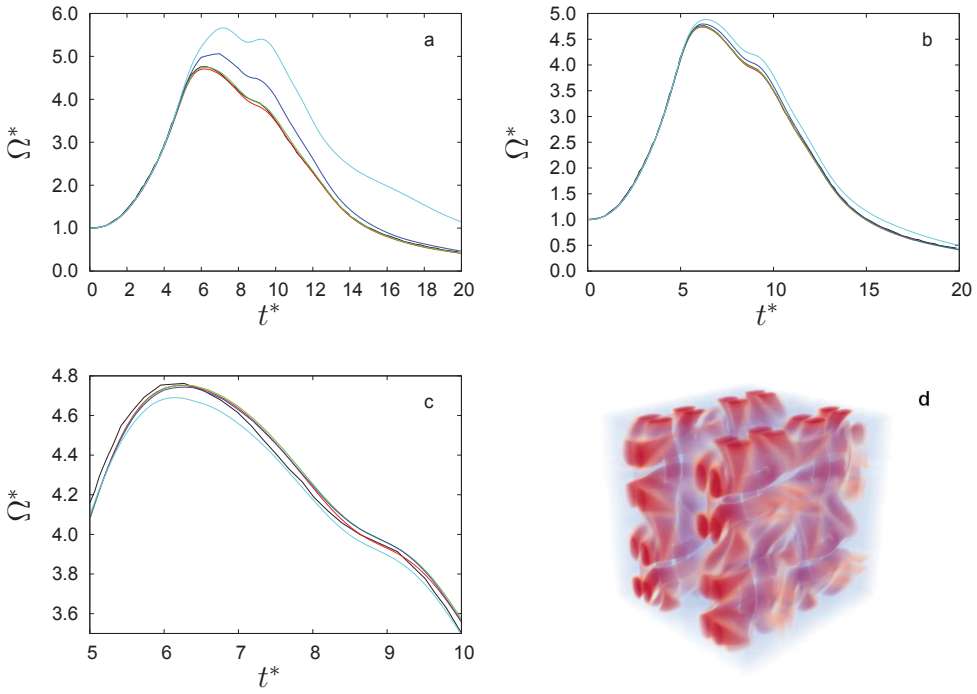


Figure 4.14: Flow simulation of Taylor-Green vortices at Re=300 with reprojection every (red), every 10th (green), every 100th (blue) time step and no reprojection (cyan) at (a) $128 \times 128 \times 128$ and (b) $256 \times 256 \times 256$ resolution. (c) Enstrophy at $128 \times 128 \times 128$ resolution with varying time steps $\Delta t^* = 0.01, 0.02, 0.04, 0.08$ (red,green,blue,cyan). (d) Volume rendering of the magnitude of vorticity at $t^* = 6$ from no vorticity (transparent blue) to 25% of maximum vorticity magnitude (solid red).

50% entailing an increase of the kinetic energy (not shown). Increasing the reprojecting rate improves the solution and provides accurate results for reprojection done each or every 10th time step. As expected divergent numerical errors are reduced by increasing the resolution to $256 \times 256 \times 256$. Without reprojection the agreement to [11] is better for $N_x = 256$ than reprojection every 100 time steps at $N_x = 128$ resolution. It is noteworthy that reprojection at proper intervals reduces the requirements to the numerical resolution.

The simulation of the Taylor-Green vortices depends on the size of the time step Δt^*. For $N_x = 128$ there is good agreement for $\Delta t^* = 0.08$, close to the stability criterion of the diffusion term. Varying the time discretisation causes the strongest deviation of the enstrophy at $t^* = 6$. Visualising the vorticity at $t^* = 6$ shows a heavily stretched vorticity field, Fig. 4.14d,

compared to the initialised vorticity, see Fig. G.2, Appendix G.

4.3 Solid boundaries

Imposing solid boundaries using Brinkman penalization [1, 50, 20] is done analogously to two-dimensional flow, section 3.2. With the penalization term the vorticity equation becomes

$$\frac{D\omega}{Dt} = \nu\boldsymbol{\nabla}^2\omega + \boldsymbol{\nabla}\times\left[\lambda\chi(\boldsymbol{u}_s - \boldsymbol{u})\right], \tag{4.3.1}$$

where λ is the penalization parameter and $\boldsymbol{u}_s$ is the solid velocity. χ is the solid mask that varies smoothly from 1 inside the solid to 0 in the fluid. This *mollification interval* has the width of a fixed factor of cell diagonal ϵ. The solid is thus included in the fluid domain while the flow field converges to the solid boundaries/velocity [1]. The mollification interval converges $\mathcal{O}(1)$ with the resolution. Regions of high curvature or kinks will be particularly sensitive to the resolution. This makes the definition of the boundary less precise and dependent on the variation of χ within the mollification interval. In the present and in former [82] work it has been seen that the mollification interval introduces *penalization blockage* resembling the concept of displacement thickness. By offsetting the mollification interval into the solid by a fixed multiple of ϵ the blockage may be reduced while the mollification converges with the resolution. By imposing penalization as a right hand side contribution the temporal convergence is expected to follow the order of the time integration.

For explicit integration the penalization parameter is restricted by the time step $\lambda \leq \frac{1}{\Delta t}$ to ensure numerical stability [50]. The penalization is made unconditionally stable by evaluation in a split-step algorithm [20]. The vorticity is replaced by an intermediate value according to the penalization term

$$\tilde{\boldsymbol{u}}^{n+1} = \frac{\boldsymbol{u}^n + \lambda\Delta t\chi\boldsymbol{u}_s^n}{1+\lambda\Delta t\chi} \tag{4.3.2}$$

$$\tilde{\boldsymbol{\omega}}^{n+1} = \boldsymbol{\nabla}\times\tilde{\boldsymbol{u}}^{n+1}. \tag{4.3.3}$$

Subsequently the stretching and diffusion terms are applied to $\tilde{\boldsymbol{\omega}}^{n+1}$.

This implies, as discussed in section 3.2, convergence off $\boldsymbol{u}_s$, steep gradients and significant numerical diffusion. The latter has been observed in section 4.2.4 (Fig. 4.13b). Instead the vorticity is updated according to the penalization

$$\Delta\boldsymbol{u} = \chi(\boldsymbol{u}_s^n - \boldsymbol{u}^n) \tag{4.3.4}$$

$$\Delta\boldsymbol{\omega} = \boldsymbol{\nabla} \times \Delta\boldsymbol{u} \tag{4.3.5}$$
$$\tilde{\boldsymbol{\omega}}^{n+1} = \boldsymbol{\omega}^n + \Delta\boldsymbol{\omega}. \tag{4.3.6}$$

which does not modify vorticity in the fluid domain and allows full control of the mollification interval.

The aerodynamic force on the solid equals the force on the fluid with opposite sign. It can therefore be computed by integrating the penalization term in Eq. (4.3.1) or the time averaged velocity update

$$\boldsymbol{F} = \frac{\Delta\boldsymbol{u}}{\Delta t} = \frac{\chi(\boldsymbol{u}_s^n - \boldsymbol{u}^n)}{\Delta t}, \tag{4.3.7}$$

where Δt is the length of the time step. Alternate methods for computing the aerodynamic forces are the time derivative of the total vorticity moments [115] in an unbounded domain or in a sub region of the domain [70]. Furthermore, the solid force can be computed by integrating the pressure drop through the domain when the boundaries are periodic [82].

4.3.1 Spheres in a simple cubic array

Hill has investigated the steady drag on spheres in periodic arrays [44]. Presently a simple cubic array with a solid volume fraction 0.2 is considered corresponding to the ratio $D/L = 0.726$ where D is the sphere diameter and L is side length of the cubic domain. The domain is discretised evenly by N_x points in all coordinate directions. The Reynolds number is defined as

$$\mathrm{Re}_D = \frac{\bar{U}D}{\nu} = 204, \tag{4.3.8}$$

where $\bar{U}$ is the average velocity across the periodic domain.

Hill non-dimensionalises stationary force results with the stokes flow drag force $3\pi\nu D\bar{U}/\rho$, ρ is the fluid density. In the present work all results are non-dimensionalised as the coefficient of drag

$$C_D = \frac{F_D}{\frac{1}{2}\rho\bar{U}^2\frac{\pi}{4}D^2} \tag{4.3.9}$$

and time is non-dimensionalised as

$$t^* = t\frac{\bar{U}}{D}. \tag{4.3.10}$$

The drag force during the onset flow is shown in Fig. 4.15a for various resolutions ($N = 64, 128$) and mollification intervals ($1\epsilon, 2\epsilon$). The time

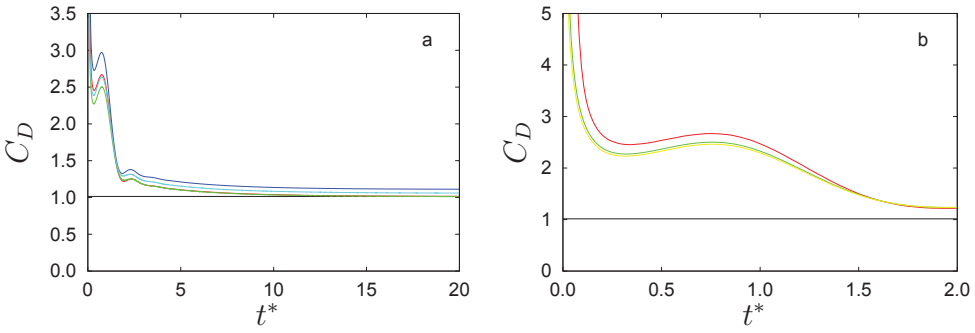


Figure 4.15: Coefficient of drag on a sphere in a simple cubic array at Re=102 (left). For a 1ϵ mollification interval at resolution $N = 64$ (red), $N = 128$ (green), $N = 256$ (yellow) and a 2ϵ mollification interval at $N = 64$ (blue), $N = 128$ (cyan). The close-up (right) shows the drag to be almost converged with $N = 128$ and a 1ϵ mollification interval.

integration is done by discrete time steps based on the stability criterion Eq. (4.1.4). Respectively $\Delta t^* = 0.008$ and $\Delta t^* = 0.002$ for $N = 64$ and $N = 128$. Vorticity reprojection is performed every 10 time steps. Due to penalization blockage the drag force is generally higher the lower the resolution and the wider the mollification interval. The dependence on the numerical parameter is highest for $t^* < 2$ when the drag force fluctuates. For a 1ϵ penalization interval the converged drag deviates from the reference [44] by 0.08% and 0.24% for $N = 64, 128$ respectively. Fig. 4.15b shows a close up of the initial drag. For $N = 128$ the drag after the initial singular force deviates 2% from the drag computed at $N = 256$.

A 1ϵ wide mollification interval is considerably thinner than the $\mathcal{O}(4)$ finite difference curl used to apply the penalization. In the periodic array strong viscous forces ensure a thick and smooth boundary layer around the sphere, see Fig. 4.16, thereby relaxing the requirements to the width of the mollification interval. More visualisations of this flow can be seen on Fig. G.3 in Appendix G.

4.3.2 Spheres in free-space flow

The initial onset flow on spheres in a free-space flow has been studied at various Reynolds numbers by Dennis [31] using an axisymmetric series expansion.

The domain is discretised to encompass the entire vorticity field during the simulation and is resolved corresponding to N_D grid points per sphere

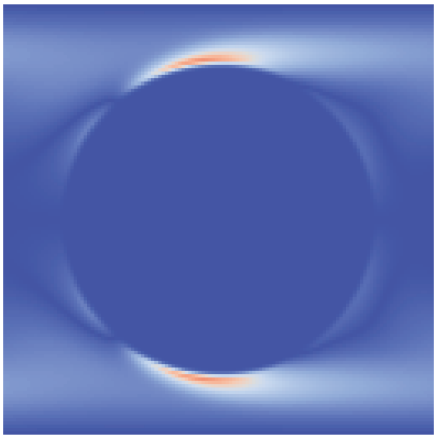


Figure 4.16: A plane intersecting the centre of a sphere in a periodic array clearly outlined by the (magnitude of) boundary layer vorticity in a steady state. Blue corresponds to no vorticity and red corresponds to the maximum magnitude of vorticity. The flow moves from left to right.

diameter. The velocity is computed by the spectral curl of the stream function implying the use of G, not $\boldsymbol{K}$, thus allowing solution of the Poisson associated with vorticity reprojection. Eq. (4.2.16) is used for the singularity of the free-space Green's function. A wider 3ϵ mollification interval is initialised as numerical divergence of the vorticity field otherwise becomes significant.

Fig. 4.17a shows the drag coefficient following the impulsively started flow for Re $= 40$ the drag coefficient following the impulsively started flow. For $\Delta t^* = 0.001$ and $N_D = 21$ the drag force is significantly higher than the reference [31] due to the mollification interval and the low resolution. By offsetting the mollification interval 1ϵ into the solid the converged drag does not deviates beyond the digitisation error of the reference. By increasing the resolution to $N_D = 43$ and using no offset of the mollification interval the deviation to the reference is improved from 25.6% to 8.8%. The deviation from $N_D = 43$ resolution is improved to a deviation 3.3% below the reference by offsetting the mollification interval 1ϵ.

For Re $= 100$ Fig. 4.17b shows the drag computed with a 1ϵ offset of the mollification interval. For $N_D = 21$ and $N_D = 43$ the deviation to the converged drag is 5.107% and 0.329% respectively. Without the mollification offset these deviations amount to 33.85% and 13.89% respectively.

The presented results are obtained from simulations using a fixed time step $\Delta t^* = 0.001$. The primary effect of increasing the size of the time

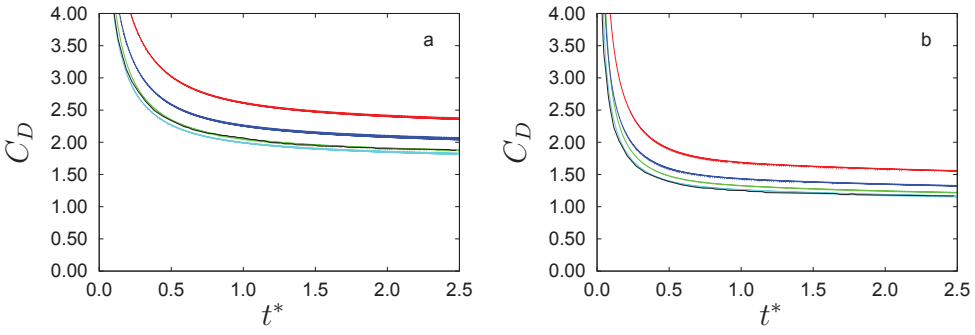


Figure 4.17: Coefficient of drag on a sphere in an impulsively started flow for (a) Re=40, (b) Re=100. For $N_D = 21$ with a 3ϵ interval without offset (red) and offset by 1ϵ (green). For $N_D = 43$ with a 3ϵ interval without offset (blue) and offset by 1ϵ (cyan).

step is a greater deviation of the course of the initial drag to the reference whereas the conclusions with respect to the converged drag do not change (not shown).

As the Reynolds number is increased the flow converges faster to a steady state solution. The assumption of axisymmetry by the reference [31] can still be expected to hold at Re = 200, but for Reynolds numbers above $\mathrm{Re}_1 \approx 210$–$212$ numerical, analytic and experimental studies show that the flow becomes non-axisymmetric [77]. Fig. 4.18a shows C_D for $N_D = 21, 43, 85$ converging well to the reference until $t^* = 0.2$ after which a substantial deviation is observed. For the steady drag force the $N_D = 21$ and $N_D = 43$ resolutions deviate from the $N_D = 85$ resolution by 9.8% and 2.5% respectively. The steady drag force computed with $N_D = 85$ resolution deviates from the reference by 8.0% indicating that the simulated results do not converge towards the reference and that the flow is resolved well at $N_D = 43$. This is supported by the flow visualisation for $N_D = 43$, Fig. 4.18b. Experimental results of the drag force have been fitted [109] by the curve

$$C_D \approx \frac{24}{\mathrm{Re}} + \frac{6}{1 + \sqrt{\mathrm{Re}}} + 0.4, \tag{4.3.11}$$

valid for Reynolds number in the range from 0 to 10^5. Eq. (4.3.11) evaluates to $C_{D,200} = 0.9162$ at Re = 200. The reference and the present results deviate from $C_{D,200}$ by 8.9% and 1.6%.

Good agreement with the reference is found for $Re = 40, 100$ and all the present results are self-consistent. As $C_{D,200}$ is obtained experimentally

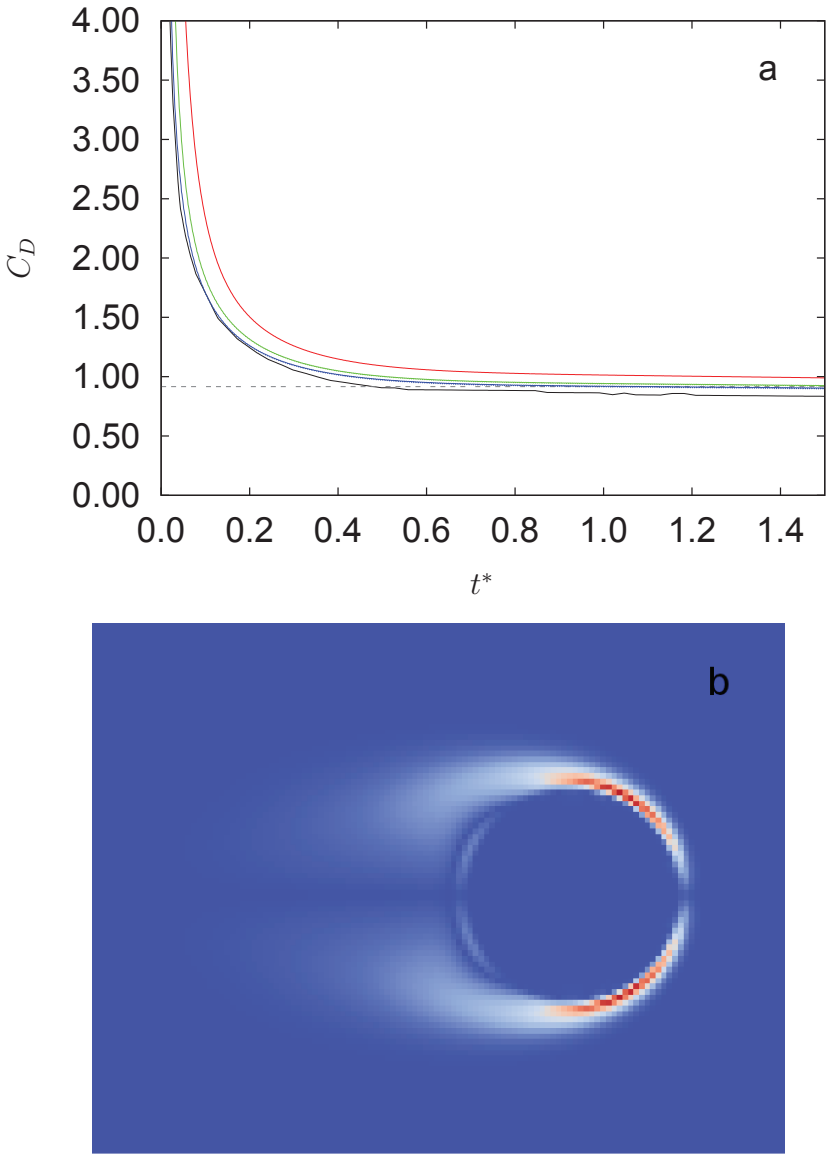


Figure 4.18: (a) Coefficient of drag on a sphere in an impulsively started flow for Re=200 simulated with an 1ϵ offset of the mollification interval. For $N_D = 21$ (red), $N_D = 43$ (green) and $N_D = 85$ with a 3ϵ interval. (b) Visualisation of the magnitude of vorticity at $t^* =$ for $N_D = 43$.

conclusions will be drawn carefully; the present results for Re = 200 show fair agreement with the experimental reference and good correspondence with [31] for the initial development of the drag.

4.4 Three-dimensional multi resolution VIC using PPM

The need for parallel computations easily arises when doing three-dimensional time-dependent flow simulations. The simulations become computationally expensive with regard to processing power as well as memory. By distributing the computational load among several processors with a greater total memory, larger problems may be treated and at shorter simulation time due to the increased computing power. The processors may either be in a multi-core machine with shared memory or on separate computational nodes, referred to as distributed memory. Shared memory allows the use of OpenMP which offers easy but less transparent parallelisation of existing code. Distributed memory on the other hand requires network data exchange and is more laborious but allows greater control of the parallel process. Though the performance of shared memory systems is growing the greatest high performance computing power[1] is available, by a very great margin, on distributed systems. However, within each computational node shared memory is often utilised. The complexity of structuring a parallel program can be greatly reduced through the use of software libraries. The Parallel Particle Mesh library [92] (PPM) provides routines for domain decomposition, data exchange between computational nodes, particle-, mesh- and particle/mesh-operations among other things. A program using (linking to) the PPM library functionality is called a PPM *client*. However, to date PPM does not have an inherent multiresolution structure. Any such structure requires consideration of data storage, information exchange between resolution levels as well as parallel computational efficiency.

An important PPM term is the *topology* [92]. A topology is a parallel decomposition of the computational domain into *subdomains* and topologies may contain particle data and/or mesh data. A *mesh* is an Eulerian decomposition of the domain and several data sets, *fields*, may be discretised according to the mesh. PPM provides all necessary functions for interpolation between meshes and particles. Several topologies may exist for the same domain (e.g. decomposition into Cartesian blocks, pencils, slabs, etc.).

[1] www.top500.org

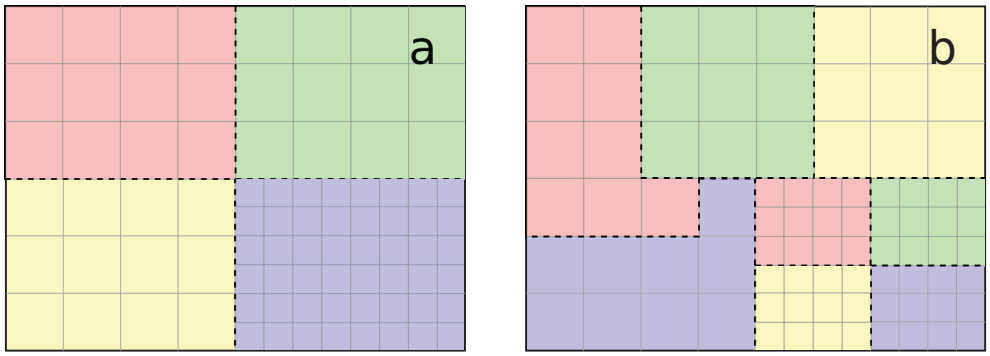


Figure 4.19: Decomposition of a domain with increased resolution in the lower right corner. (a) For a global Cartesian decomposition the blue node is the most strongly loaded. (b) An even composition allows even loading of the nodes while ensuring inter-resolution communication within each node.

Each subdomain of a topology is assigned to a *processor* (computational node) and each processor may have been assigned several subdomains. The subdomains may exchange data with their neighbouring subdomains which is referred to as a *local mapping*. Or all data may have to be reassigned to another topology, called a *global mapping* as all processors potentially communicate with each other.

This poses questions regarding the decomposition of data in multiple resolution levels. Adaptivity using patches as defined in sections 3.3, 3.4 and 3.5 is localised geometrically and requires communication between parent and patch.

By using one global domain decomposition on all resolution levels network communication would be minimised as the patch and parent segments would be stored on the same processor. However this data distribution easily entails an imbalance of the computational load. The dependency between parents and patches in the algorithm section 3.5 requires that resolution levels are traversed simultaneously. The result is low parallel efficiency due to idle processors if one simple decomposition is used on all resolution levels, such as Cartesian blocks between end-to-end planes (Fig. 4.19a).

This may be avoided, if the number of processors does not greatly exceed the computational requirements at any level. The decomposition should not be restricted to planes intersecting the entire domain but should instead allow individual subdivision of regions covered by patches; at all resolution levels. Subsequently the unassigned regions of the parent level must be decomposed and so forth, see Fig. 4.19b. Hereby the computational load is equally distributed and data exchange between resolution levels is kept

within each processor. The drawback of this decomposition is a larger parallel overhead due to higher surface to volume ratio of the increased number of subdomains. Furthermore the size of the patch and the resolution of the underlying base patch P_1^1 may restrict both positioning and the number of processors in the simulation. Thus PPM must be extended such that not all processors partake in each VIC problem/patch to ensure parallel efficiency.

If the parallel overhead associated with the subdomain boundaries is ignored the subdivision can be done such that the load of all processors will be approximately equal. If not, the optimum is a compromise between keeping the processors busy and minimising both local and global mappings and may not be easily found. Several decomposition tools exists [49, 33] but the decomposition has not been further studied in the present work.

4.4.1 PPM multiresolution data structure

Three approaches have been considered for the multiresolution data structure, requiring varying degrees of modification to PPM:

- **Per-level topology**: One topology is initialised for each resolution level. This implies that there will be unfilled parts of each topology requiring increased bookkeeping that may be stored on the meshes (neither is currently part of PPM). The Poisson equation will be solved in the patch regions and not on the entire topology. This data structure leaves the bookkeeping mostly to PPM which could mean the client-side calls to the involved routines will be few and clean.

  Particles may potentially move between patches. This method leaves full control with respect to the decomposition and the discussed issues regarding network communication vs activity of the available processing power.

- **Per-domain topology**: One topology is initialised for the entire domain and all resolution levels. The approach is very similar to the method outlined above and the additionally required information may be stored on the mesh data structure. Particles would need to be either stored in separate data arrays or be marked with the resolution level to ensure interpolation to meshes of the correct resolution level.

- **Per-patch topology**: For each patch a topology is initialised. This supports the underlying principle of the multiresolution VIC algorithm, section 3.5, that each patch is an isolated problem connected to the other patches through interpolations. The interpolations and mappings between different, but overlapping, topologies are not currently

supported in PPM. This method requires minimal alteration of PPM and leaves the entire multiresolution structure to the client. Presently PPM assigns all processors to each topology. The option for more specific assignment would increase per processor load and thereby the parallel efficiency. This would also allow full control of the decomposition. As the ensemble of topologies can be regarded as a cell list particle-mesh interpolation is alleviated.

For all of the above outlined methods PPM must be extended from the current vertex-centred data format to also support cell-centred data. Currently each PPM subdomain has redundant $n + 1$ points, where n is the number of cells of a subdomain in a direction. These points have been introduced to ease P2M interpolation and can be thought of as an additional asymmetric layer of ghost cells. The $n + 1$ layers should be removed. When remeshing particles or interpolating between particles and meshes, an offset corresponding to the shift from vertex- to cell-centred data must be introduced.

The present implementation is prepared for the latter of the three methods described above. In the present work data is interpolated between resolution levels and mapped between the corresponding topologies. All mesh-to-mesh (M2M) interpolation will be called client-side and the mapping done with the updated PPM routines. As particles leave one subdomain it is transferred to the adjacent subdomain through local mappings. If a particles cannot be assigned PPM issues an error and halts. A flag has been added to the mapping API to ignore unassigned particles thereby discarding particles leaving a topology. If a particle leaves its topology during one of several Runge-Kutta sub steps it will be discarded.

Patches may be moved during the simulation by modifying the topology bounds before P2M interpolation of existing particles.

4.5 Application: Medusa

The swimming motion of a medusa, commonly known as a jellyfish, and its wake has been studied extensively in the literature, e.g. [29, 28, 26]. It serves well as a test of the capabilities of the Brinkman penalization due to the moving and deforming motion of the solid. As the penalization is a volume force the solid velocity $\boldsymbol{u}_s$ must be known within the entire penalized volume. Furthermore, the solid velocity must be solenoidal, a consequence of including the solid in the fluid domain. The simulation of the medusa motion has been implemented in the `wvic` using implicit penalization, Eq. (3.2.2).

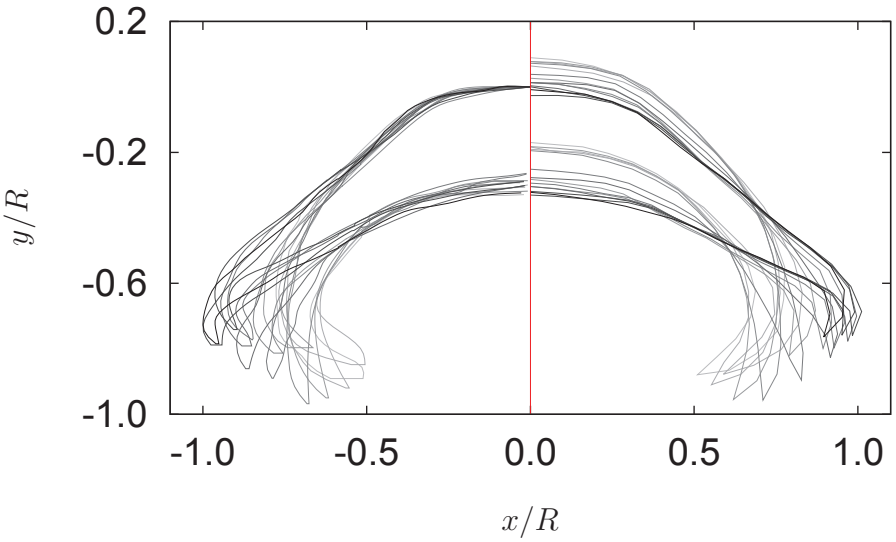


Figure 4.20: The series of outlines describing a swimming stroke of the medusa. To the left of the symmetry axis (red) the original data. To the right the volume- and surface node corrected data. The corrected data is in a frame of reference following the centre of mass.

The initialisation of the solid mask χ_s and velocity $\boldsymbol{u}_s$ is based on an existing STL-input[2] routine [82].

The input data for the simulation is a series of cross sections/frames captured from video footage of *Aurelia aurita* by courtesy of John O. Dabiri[3]. 29 consecutive outlines of the medusa describe two swimming strokes. The left side of Fig. 4.20 shows the frames of one swimming stroke. Under assumption of a rotationally symmetric shape of the medusa, the volumes of the frames vary up to 50%. Furthermore the irregular frames are modified into regularly spaced panels suitable for numerical initialisation of the solid. The procedure is as follows:

[2]STL: Originates from 'Stereolithography CAD' and is a file format containing a triangulated surface representation of an object through triangles described by three corner points and a normal. The STL-file contains no information on the connectivity of the discrete elements is.

[3]John O. Dabiri, Professor of Aeronautics & Bioengineering, California Institute of Technology

- The nodes describing the surface are reinitialised with a constant spacing on the upper surface and a constant spacing on the lower surface. The upper and lower surfaces are discretised with the same number of points. The surfaces share the apex point and both surfaces have a point lying on the symmetry axis, thereby closing the geometry.

- A few frames are substituted by the average of the preceding and succeeding frames due to extreme deviation from the series.

- The midpoint line (plane) is determined between corresponding nodes on the upper and lower surfaces. The medusa is divided into subvolumes between two upper and two lower surface nodes and two corresponding midpoint nodes, i.e. subvolumes both above and below the midpoint line. The average volume distribution of the series is found as a reference. This is done in a cylindrical frame of reference.

- The effect of the error from discretising the video footage is assumed to vanish when averaging the volume distribution of the frames. The surface points of all frames are corrected such that the total volume and the volume distribution along the medusa dome is constant. As the position of the midpoint line is based on the position of the two surfaces it will be more accurate than either of the two surfaces.

- Once more the surface points are redistributed equally and coordinates of the nodes are shifted to that of the centre of mass.

- The velocity of the surface is computed in each node from a central difference approximation of the first derivative in time.

As the simulation progresses $\chi(t)$ must, each time step, be initialised to the new shape at time t . Each frame corresponds to a discrete time and therefore the shape and velocity of the solid (medusa) is interpolated linearly to time t. Based on the cylindrical coordinates of each discrete point in the domain χ and $\boldsymbol{u}_s$ is determined from bi-linear interpolation within the sub-cells.

During the simulation the force on the medusa is calculated and the acceleration found, assuming ambient density of the medusa. The velocity is computed by integrating the acceleration and the position of the centre of mass of the medusa is in turn found by integrating the velocity. As the movement of the medusa is given from the series of outlines there is no backcoupling from the flow to the shape of the solid. Fig. 4.21 shows vorticity in the flow after ten strokes of the medusa. In the immediate wake of the medusa vortex rings shed by the swimming motion can be seen. Further downstream the starting vortex is still visible and stable. During each stroke

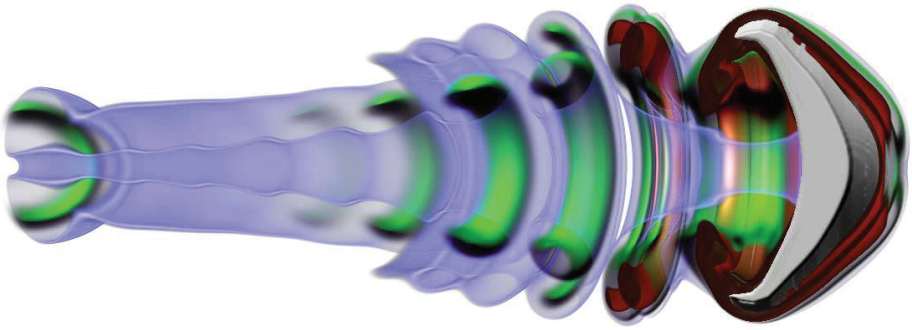


Figure 4.21: Volume rendering of magnitude of vorticity after ten swimming strokes of the medusa. Rendering by Diego Rossinelli. The Reynolds number is $D^2/T\nu = 5000$.

the medusa sheds a starting vortex that is shed as the bell contracts. Then a stopping vortex is formed as the bell re-expands. The stopping vortex is expelled along side the starting vortex during the following contraction [85]. As the medusa starts moving the centre of mass obtains a periodically steady motion after 8 strokes, see Fig. 4.22a. The periodic motion is clear from the velocity of the medusa. Initially the velocity assumes negative values during parts of the stroke but with a net forward motion. Most probably this is not common behaviour of the medusa but is instead likely to be due to the fixed motion of the frames. For a live medusa back coupling may consist both of the fluid forcing as well as modified behaviour due whatever sensory organs the medusa possess. This is conjectured to affect the actuation of the medusa during the initial motion, as well as the reexpansion of the bell. In particular the latter, as reexpansion of the medusa bell must be done by fluid forces (and possibly interior elastic forces) due to the medusa physiology [27]. With the prescribed deformation of the medusa a potentially strong forcing may appear during reexpansion. The time discrete nature and unsteadiness of the frames show peaking forces and accelerations, see Fig. 4.22b, making the piecewise motion of the original frames apparent. A live medusa may not be able to produce the peak accelerations whereas the average magnitude of acceleration is just 20% of the peak accelerations.

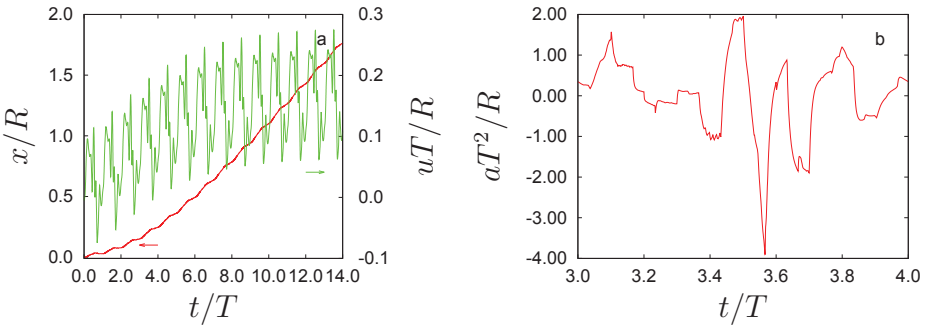


Figure 4.22: (a) Position (red) and velocity (green) of the medusa centre of mass as a function of time. (b) The acceleration (red) of the centre of mass during a single stroke. The Reynolds number is $D^2/T\nu = 5000$.

Figure 4.23: Flow around a section of the approach span of the Great Belt bridge at Re=1000. Magnitude of vorticity from blue to red (minimum to maximum) visualised at $tU/C = 7.5$.

4.6 Application: Bridge section

The flow around a finite section of the approach span to the Great Belt East bridge has been simulated, corresponding to the bridge during the construction phase. The length L of the section is 4 chords C. The geometry is triangulated into the STL format and initialised in `naga` using an STL-initialisation routine [82] adapted to the new implementation `naga`. Fig. 4.23 shows the initialised structure. The solid is placed $0.25C$ off of the upwind face of the $10C \times 6C \times 6C$ domain and centred horizontally as well as vertically. The domain is resolved by $N_C = 32$ cells/particles per chord length. The Reynolds number is

$$\mathrm{Re}_C = \frac{UC}{\nu} = 1000 \tag{4.6.1}$$

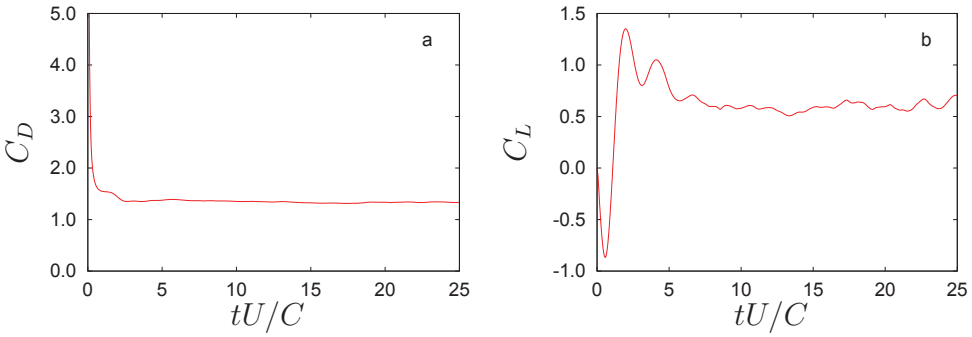


Figure 4.24: The aerodynamic (a) lift and (b) drag on a section of the approach span of the Great Belt East bridge at Re = 1000.

to reasonably resolve the flow structures. The aerodynamic drag F_D and lift F_L forces are non-dimensionalised as

$$C_D = \frac{F_D}{\frac{1}{2}\rho U^2 H L} \tag{4.6.2}$$

$$C_L = \frac{F_L}{\frac{1}{2}\rho U^2 H L} \tag{4.6.3}$$

respectively, where ρ is the fluid density and $H = 0.28C$ the section height.

The drag force on the bridge peaks at the onset flow but after $tU/C = 5$ assumes a constant value $C_D = 1.33 \pm 0.03$, see Fig. 4.24a. Initially the lift force fluctuates smoothly but does not become harmonic as expected [61], see Fig. 4.24b. At $tU/C = 7.5$ the oscillations appear random and visualising the flow shows a an immediate wake consisting of complex vortex structures, see Fig. G.4 in the appendix. The lift signal becomes disharmonic before the starting vortex reaches the downstream end of the domain suggesting that the development is not causes by vorticity leaving the domain. The flow initially develops parallel alternating vortex lines. Streamwise vortices generated at the ends of the section cause the loops of parallel vortices to contract. Eventually both the immediate and the far-field wake becomes turbulent without clear vortical structures parallel to the bridge.

The three-dimensional flow around actual bridge decks is expected to generate harmonic oscillations of the lift force. The flow around the finite bridge section is strongly influenced by end effects from the finite bridge section. The lift force can be expected to be a harmonic signal for a very long bridge section or by imposing a periodic boundary condition [10] along the direction of the bridge span.

4.7 Summary

Several solutions have been presented for grid based numerical solution to the Poisson equation using the singular Green's function, including substitution of the singularity by finite values and regularisation techniques known from vortex particle methods. Using the framework of the implemented Poisson solver high order error convergence has been shown.

The vortex-in-cell implementation is validate against periodic and free-space flow around spheres. The versatility of the VIC algorithm coupled with Brinkman penalization is demonstrated and investigated.

Conclusions

The work presented in this dissertation presents major contributions to vortex particle methods - both in terms of methods extending its application and efficiency. The work is expected to be relevant to both the industry and academics with relevance to areas ranging from fluid mechanics over electro-, molecular- or plasma dynamics to celestial mechanics.

A new technique for the calculation of the aerodynamic admittance in bluff body aerodynamics is presented. The method is based on the two-dimensional discrete vortex method and introduces a turbulent oncoming flow through the insertion of upstream vortex particles modelling the anisotropic turbulent velocity spectra. The admittances of the lift and pitching moment are obtained from the measured spectra of the turbulent flow field and the corresponding spectra of the aerodynamic loads. The method has been validated through detailed simulations of the turbulent flow past a flat plate and past the Great Belt East bridge, the Øresund bridge and the Busan-Geoje bridge. The results were found in good agreement with the semi-analytical model of Liepmann and wind tunnel experiments. Suggestions to improvement of the method are higher order grid-to-particle conversion and non-uniform particle seeding.

A novel multiresolution vortex-in-cell (VIC) algorithm using patches of varying resolution is presented. The algorithm is based on free-space FFTs for the solution of the Poisson equation on the patches. Superposition is used to compute the contribution from vorticity of separate meshes. The present multiresolution algorithm is implemented in two dimensions but is, like the general VIC algorithm [9], not restricted to two dimensions. It is demonstrated that penalization can be carried out as an interpolation between the solid and fluid velocities in a manner similar to the semi-implicit penalization method. This interpolating penalization eliminates the penalization parameter and yields increased control of the mask defining the solid. The algorithm is validated by considering the impulsively started flow past fixed and rotating circular cylinders. The flow past a suspension bridge including railings and crash barriers demonstrate the ability of the proposed

multiresolution VIC algorithms to handle complex geometries.

An existing three-dimensional VIC implementation is extended to impose the no-slip boundary condition of deformable and moving objects thereby demonstrating the versatility of Brinkman penalization. This is exemplified by three-dimensional simulation of the flow around a swimming medusa. A new Fourier based Poisson solver for the parallel framework, PPM, is introduced and validated in conjuction with a new PPM free-space vortex-in-cell implementation. Multiresolution data structures for the PPM library are discussed. The numerical effects of the singular Green's function are investigated and high order error convergence is obtained from novel use of regularisation techniques.

Bibliography

[1] P. Angot, C.-H. Bruneau, and P. Fabrie. A penalization method to take into account obstacles in incompressible viscous flows. *Numer. Math.*, 81:497–520, 1999.

[2] E. Arquis and J.-P. Caltagirone. On the hydrodynamical boundary-conditions along a fluid layer porous-medium interface - application to the case of free-convection. *C. R. Acad. Sci. II*, 229(1):1–4, 1984.

[3] J. Barnes and P. Hut. A hierarchical $O(N \log N)$ force-calculation algorithm. *Nature*, 324(4):446–449, 1986.

[4] G. Barton. *Elements of Green's Functions and Progagation. Potentials, Diffusion and Waves*. Oxford University Press, 1989.

[5] J. T. Beale and A. Majda. High order accurate vortex methods with explicit velocity kernels. *J. Comput. Phys.*, 58:188–208, 1985.

[6] M. Bergdorf, G.-H. Cottet, and P. Koumoutsakos. Multilevel adaptive particle methods for convection-diffusion equations. *Multiscale Model. Simul.*, 4(1):328–357, 2005.

[7] C. K. Birdsall and D. Fuss. Clouds-in-clouds, clouds-in-cells physics for many-body plasma simulation. *J. Comput. Phys.*, 3:494–511, 1969.

[8] J. Carrier, L. Greengard, and V. Rokhlin. A fast adaptive multipole algorithm for particle simulations. *SIAM J. Sci. Stat. Comput.*, 9(4):669–686, 1988.

[9] P. Chatelain, A. Curioni, M. Bergdorf, D. Rossinelli, W. Andreoni, and P. Koumoutsakos. Billion vortex particle direct numerical simulations of aircraft wakes. *Comp. Meth. Appl. Mech. & Engng.*, 197:1296–1304, 2008.

[10] P. Chatelain and P. Koumoutsakos. A Fourier-based elliptic solver for vortical flows with periodic and unbounded directions. *J. Comput. Phys.*, 229:2425–2431, 2010.

[11] Y. Chen and V. Rokhlin. On the inverse scattering problem for the Helmholtz equation in one dimension. *Inverse Problems*, 8:365–391, 1992.

[12] H. Cheng, L. Greengard, and V. Rokhlin. A fast adaptive multipole algorithm in three dimensions. *J. Comput. Phys.*, 155:468–498, 1999.

[13] A. J. Chorin. Numerical study of slightly viscous flow. *J. Fluid Mech.*, 57(4):785–796, 1973.

[14] M.-H. Chou. Numerical study of vortex shedding from a rotating cylinder immersed in a uniform flow field. *Int. J. Numer. Meth. Fluids*, 32(5):545–567, 2000.

[15] O. Christensen. *Differentialligninger og uendelige rækker*. Institut for matematik, Technical University of Denmark, 2006.

[16] O. Christensen and K. L. Christensen. *Approximation Theory: From Taylor Polynomials to Wavelets*. Birkhäuser Boston, 2004.

[17] J. P. Christiansen. Numerical simulation of hydrodynamics by the method of point vortices. *J. Comput. Phys.*, 13:363–379, 1973.

[18] G. Clementson. *An Investigation of the Power Spectral Density of Atmospheric Turbulence*. D.Sc. thesis, Massachusetts Institute of Technology, 1950.

[19] R. Cocle, G. Winckelmans, and G. Daeninck. Combining the vortex-in-cell and parallel fast multipole methods for efficient domain decomposition simulations. *J. Comput. Phys.*, 227:9091–9120, 2008.

[20] M. Coquerelle and G.-H. Cottet. A vortex level set method for the two-way coupling of an incompressible fluid with colliding rigid bodies. *J. Comput. Phys.*, 227(21):9121–9137, 2008.

[21] G. H. Cottet. A particle-grid superposition method for the Navier-Stokes equations. *J. Comput. Phys.*, 89:301–318, 1990.

[22] G.-H. Cottet. Particle-grid domain decomposition methods for the Navier-Stokes equations in exterior domains. *Lectures Appl. Math.*, 28:103–117, 1991.

[23] G.-H. Cottet and P. Koumoutsakos. *Vortex Methods – Theory and Practice*. Cambridge University Press, New York, 2000.

[24] G.-H. Cottet, P. Koumoutsakos, and M. L. O. Salihi. Vortex methods with spatially varying cores. *J. Comput. Phys.*, 162(1):164–185, 2000.

[25] G.-H. Cottet and P. Poncet. Advances in direct numerical simulation of 3D wall-bounded flows by vortex-in-cell methods. *J. Comput. Phys.*, 193:136–158, 2003.

[26] J. O. Dabiri. Optimal vortex formation as a unifying principle in biological propulsion. *Annu. Rev. Fluid Mech.*, 41:17–33, 2009.

[27] J. O. Dabiri, S. P. Colin, and J. H. Costello. Morphological diversity of medusan lineages constrained by animal-fluid interactions. *J. Exp. Biol.*, 210(11):1868–1873, 2007.

[28] J. O. Dabiri, S. P. Colin, J. H. Costello, and M. Gharib. Flow patterns generated by oblate medusan jellyfish: field measurements and laboratory analyses. *J. Exp. Biol.*, 208(7):1257–1265, 2005.

[29] J. O. Dabiri and M. Gharib. Sensitivity analysis of kinematic approximations in dynamic medusan swimming models. *J. Exp. Biol.*, 206(20):3675–3680, 2003.

[30] A. G. Davenport. The dependence of wind load upon meteorological parameters. *Proceedings of the International Research Seminar on Wind Effects on Buildings and Structures*, pages 19–82, 1968.

[31] S. Dennis and J. Walker. Numerical solutions for time-dependent flow past an impulsively started sphere. *Phys. Fluids*, 15(4):517–525, 1972.

[32] G. Deodatis. Simulation of ergodic multivariate stochastic processes. *ASCE J. of Eng. Mech.*, 122(8):778–787, 1996.

[33] D. Durand, R. Jain, and D. Tseytlin. Parallel I/O scheduling using randomized, distributed edge coloring algorithms. *J. Parallel Distrib. Comput.*, 63:611–618, 2003.

[34] M. El Ossmani and P. Poncet. Efficiency of multiscale hybrid grid-particle vortex methods. *Multiscale Model. Simul.*, 8(5):1671–1690, 2010.

[35] F. B. Farquharson. Aerodynamic stability of suspension bridges. Technical report, University of Washington Experiment Station, 1952. No. 116 Part I and III.

[36] M. Fellhauer, P. Kroupa, H. Baumgardt, R. Bien, C. M. Boily, R. Suprzem, and N. Wassmer. SUPERBOX – an efficient code for collisionless galactic dynamics. *New Astro.*, 5:305–326, 2000.

[37] Y. C. Fung. *An Introduction to the Theory of Aeroelasticity.* John Wiley & Sons, 1955.

[38] Y. J. Ge and H. F. Xiang. Computational models and methods for aerodynamic flutter of long-span bridges. *J. Wind Eng. Ind. Aerodyn.*, 96:1912–1924, 2008.

[39] D. Goldstein, R. Handler, and L. Sirovich. Modeling a no-slip flow boundary with an external force field. *J. Comput. Phys.*, 105:354–366, 1993.

[40] L. Greengard and V. Rokhlin. A fast algorithm for particle simulations. *J. Comput. Phys.*, 73:325–348, 1987.

[41] F. J. Harris. On the use of windows for harmonic analysis with the discrete Fourier transform. *Proc. IEEE*, 66(1):51–83, 1978.

[42] R. I. Harris. Some further thoughts on the spectrum of gustiness in strong winds. *J. Wind Eng. Ind. Aerodyn.*, 33:461–477, 1990.

[43] M. M. Hejlesen, J. T. Rasmussen, A. Larsen, and J. H. Walther. Aerodynamic admittance of five bridge sections - a numerical study. *J. Wind Eng. Ind. Aerodyn.*, 2012. in preparation.

[44] R. J. Hill, D. L. Koch, and A. J. C. Ladd. Moderate-Rynolds-number flows in ordered and random arrays of spheres. *J. Fluid Mech.*, 448:243–278, 2001.

[45] R. W. Hockney. The potential calculation and some applications. *Methods Comput. Phys.*, 9:136–210, 1970.

[46] R. W. Hockney and J. W. Eastwood. *Computer Simulation Using Particles.* Institute of Physics Publishing, Bristol, PA, USA, 2. edition, 1988.

[47] P. Irwin. The role of wind tunnel modelling in the prediction of wind effects on bridges. In A. Larsen and S. Esdahl, editors, *Bridge Aerodynamics. Proceeding of the International Symposium on Advances in Bridge Aerodynamics, Copenhagen, Denmark*, pages 99–117. A. A. Balkema, Rotterdam, Brookfield, 1998.

[48] M. Ito, Y. Fujino, T. Miyata, and N. Narita. *Cable-Stayed Bridges-Recent Developments and their Future*. Elsevier, 1991.

[49] G. Karypis and V. Kumar. A parallel algorithm for multilevel graph partitioning and sparse matrix ordering. *J. Par. Dist. Comp.*, 48(1):71–95, 1998.

[50] N. K.-R. Kevlahan and J.-M. Ghidaglia. Computation of turbulent flow past an array of cylinders using a spectral method with Brinkman penalization. *Eur. J. Mech., B*, 20:333–350, 2001.

[51] P. Koumoutsakos. Inviscid axisymmetrization of an elliptical vortex ring. *J. Comput. Phys.*, 138:821–857, 1997.

[52] P. Koumoutsakos and A. Leonard. High-resolution simulation of the flow around an impulsively started cylinder using vortex methods. *J. Fluid Mech.*, 296:1–38, 1995.

[53] C. Lanczos. *Applied Analysis*. Prentice Hall, 1956.

[54] G. Larose. *The dynamic action of gusty winds on long-span bridges*. Ph.D. thesis, Department of Civil Engineering. Technical University of Denmark, April 1997.

[55] G. L. Larose. Experimental determination of the aerodynamic admittance of a bridge deck segment. *J. Fluids and Structures*, 13:1029–1040, 1999.

[56] G. L. Larose and F. M. Livesey. Performance of streamlined bridge decks in relation to the aerodynamics of a flat plate. *J. Wind Eng. Ind. Aerodyn.*, 71:851–860, 1997.

[57] G. L. Larose and J. Mann. Gust loading on streamlined bridge decks. *J. Fluids and Structures*, 12:511–536, 1998.

[58] A. Larsen. Advances in aeroelastic analysis of suspension and cable-stayed bridges. *J. Wind Eng. Ind. Aerodyn.*, 74–76:73–90, 1998.

[59] A. Larsen. Computer simulation of wind-structure interaction in bridge aerodynamics. *Struct. Engng. Int.*, 8(2):105–111, 1998.

[60] A. Larsen. Aerodynamic stability and vortex shedding excitation of suspension bridges. In *Proceedings, 4th International Conference on Advances in Structural Engineering and Mechanics (ASEM08), Jeju, Korea, May 26-28, 2008*, 2008.

[61] A. Larsen, S. Esdahl, J. E. Andersen, and T. Vejrum. Storebælt suspension bridge - vortex shedding excitation and mitigation by guide vanes. *J. Wind Eng. Ind. Aerodyn.*, 88:283–296, 2000.

[62] A. Larsen, M. Savave, A. Lafrenière, M. C. H. Hui, and S. V. Larsen. Investigation of vortex response of a twin box bridge section at high and low Reynolds numbers. *J. Wind Eng. Ind. Aerodyn.*, 96:934–944, 2008.

[63] A. Larsen and J. H. Walther. Aeroelastic analysis of bridge girder sections based on discrete vortex simulations. *J. Wind Eng. Ind. Aerodyn.*, 67–68:253–265, 1997.

[64] A. Larsen and J. H. Walther. Discrete vortex simulation of flow around five generic bridge deck sections. *J. Wind Eng. Ind. Aerodyn.*, 77–78:591–602, 1998.

[65] T. Lawson. *Wind effects on building.* Applied Science Publishers Ltd., London, 1980.

[66] H. W. Liepmann. On the application of statistical concepts to the buffeting problems. *J. Aero. Sci.*, 19(12):793–800, 1952.

[67] J. J. Monaghan. Extrapolating B splines for interpolation. *J. Comput. Phys.*, 60(2):253–262, 1985.

[68] J. J. Monaghan. Particle methods for hydrodynamics. *Comput. Phys. Rep.*, 3:71–123, 1985.

[69] G. Morgenthal and J. H. Walther. An immersed interface method for the vortex-in-cell algorithm. *Computers & Structures*, 85:712–726, 2007.

[70] F. Noca, D. Shiels, and J. D. A comparison of methods for evaluating time-dependent fluid dynamic forces on bodies using only velocity fields and their derivatives. *J. Fluids and Structures*, 13:551–578, 1999.

[71] F. Noca, D. Shiels, and D. Jeon. Measuring instantaneous fluid dynamic forces on bodies, using only velocity fields and their derivatives. *J. Fluids and Structures*, 11:345–350, 1997. Special brief note.

[72] M. L. Ould-Salihi, G.-H. Cottet, and M. El Hamraoui. Blending finite-difference and vortex methods for incompressible flow computations. *SIAM J. Sci. Comput.*, 22(5):1655–1674, 2000.

[73] S. V. Patankar. *Numerical Heat Transfer and Fluid Flow*. Hemisphere, 1980.

[74] L. A. A. Pereira, M. H. Hirata, and N. M. Filho. Wake and aerodynamics loads in multiple bodies — application to turbomachinery blade rows. *J. Wind Eng. Ind. Aerodyn.*, 92:477–491, 2004.

[75] M. Perlman. On the accuracy of vortex methods. *J. Comput. Phys.*, 59:200–223, 1985.

[76] C. Peskin. Flow patterns around heart valves: A numerical study. *J. Comput. Phys.*, 10:252–271, 1972.

[77] P. Ploumhans, G. S. Winckelmans, J. K. Salmon, A. Leonard, and M. S. Warren. Vortex methods for direct numerical simulation of three-dimensional bluff body flows: Applications to the sphere at $Re = 300$, 500 and 1000. *J. Comput. Phys.*, 178:427–463, 2002.

[78] P. Poncet. Analysis of an immersed boundary method for three-dimensional flow in vorticity formulation. *J. Comput. Phys.*, 228:7268–7288, 2009.

[79] J. Prendergast. *Simulation of unsteady 2-D wind by a vortex method*. Ph.D. thesis, Department of Engineering. University of Cambridge, 2007.

[80] J. M. Prendergast and F. A. McRobie. Simulation of 2d unsteady wind by a vortex method and application to studying bluff body flow. In *7th UK Conference on Wind Engineering*, pages 1–4, 2006.

[81] J. Qiang. A high-order fast method for computing convolution integral with smooth kernel. *Comp. Phys. Commun.*, 181:313–316, 2010.

[82] J. T. Rasmussen. A penalization immersed interface method for 3D particle vortex methods. Master's thesis, Technical University of Denmark, March 2008.

[83] J. T. Rasmussen, G.-H. Cottet, and J. H. Walther. A multiresolution remeshed particle vortex method using patches. *J. Comput. Phys.*, 230(17):6742–6755, 2011.

[84] J. T. Rasmussen, M. M. Hejlesen, A. Larsen, and J. H. Walther. Discrete vortex method simulations of the aerodynamic admittance in bridge aerodynamics. *J. Wind Eng. Ind. Aerodyn.*, 98:754–766, 2010.

[85] J. T. Rasmussen, D. Rossinelli, F. Storti, P. Koumoutsakos, and J. H. Walther. 3D simulations of self-propelled, video reconstructed jellyfish using vortex methods. In *American Physical Society, 62nd Annual Meeting of the Division of Fluid Dynamics*, Minneapolis, USA, November 2009.

[86] T. A. Reinhold, M. Brinch, and A. Damsgaard. Wind tunnel tests for the Great Belt Link. In A. Larsen, editor, *Aerodynamics of Large Bridges. Proceeding of the First International Symposium on Aerodynamics of Large Bridges, Copenhagen, Denmark*, pages 255–267. A. A. Balkema, Rotterdam, Brookfield, 1992.

[87] L. Rosenhead. The formation of vortices from a surface of discontinuity. *Proc. R. Soc. Lond. A*, 134:170–192, 1931.

[88] R. Rossi, M. Lazzari, and R. Vitaliani. Wind field simulation for structural engineering purposes. *Int. J. for Numer. Methods In Engng.*, 61(5):738–763, 2004.

[89] D. Rossinelli, M. Bergdorf, G.-H. Cottet, and P. Koumoutsakos. GPU accelerated simulations of bluff body flows using vortex particle methods. *J. Comput. Phys.*, 229(89):3316–3333, 2010.

[90] E. M. Saiki and S. Biringen. Numerical simulation of a cylinder in uniform flow: Application of a virtual boundary method. *J. Comput. Phys.*, 123:450–465, 1996.

[91] S. Sakamotoa, S. Murakamib, and A. Mochida. Numerical study on flow past 2d square cylinder by large eddy simulation: Comparison between 2d and 3d computations. *J. Wind Eng. Ind. Aerodyn.*, 50:61–68, 1993.

[92] I. F. Sbalzarini, J. H. Walther, M. Bergdorf, S. E. Hieber, E. M. Kotsalis, and P. Koumoutsakos. PPM – a highly efficient parallel particle-mesh library for the simulation of continuum systems. *J. Comput. Phys.*, 215:566–588, 2006.

[93] R. H. Scanlan. Amplitude and turbulence effects on bridge flutter derivatives. *J. Struct. Engng.*, 123(2):232–236, 1997.

[94] P. Schlatter, N. A. Adams, and L. Kleiser. A windowing method for periodic inflow/outflow boundary treatment of non-periodic flows. *J. Comput. Phys.*, 206:505–535, 2005.

[95] M. Shinozuka and C.-M. Jan. Digital simulation of random processes and its applications. *J. Sound Vibration*, 25(1):111–128, 1972.

[96] E. Simiu and R. H. Scanlan. *Wind Effects on Structures: Fundamentals and Applications to Design*. John Wiley & Sons, 1996.

[97] P. A. Smith and P. K. Stansby. Impulsively started flow around a circular cylinder by the vortex method. *J. Fluid Mech.*, 194:45–77, 1988.

[98] A. Sohankar, C. Norberg, and L. Davidson. Simulation of three-dimensional flow around a square cylinder at moderate Reynolds numbers. *Phys. Fluids*, 11(2):288–306, 1999.

[99] P. K. Stansby and A. Slaouti. Simulation of vortex shedding including blockage by the random-vortex and other methods. *Int. J. Numer. Meth. Fluids*, 7:1003–1013, 1993.

[100] E. Strømmen, E. Hjorth-Hansen, S. O. Hansen, and J. Bogunovic Jakobsen. Wind tunnel tests for the Öresund link bridge, report no. 1, high bridge - section model test. Technical report, 1996.

[101] I. J. Taylor and M. Vezza. A numerical investigation into the aerodynamic characteristics and aeroelastic stability of a footbridge. *J. Fluids and Structures*, 25:155–177, 2009.

[102] T. Theodorsen. General theory of aerodynamic instability and the mechanism of flutter. TR 496, NACA, 1935.

[103] E. S. D. Unit. Characteristics of atmospheric turbulence near the ground, part ii: single point data for strong winds (neutral atmosphere). Technical report, 1993.

[104] E. S. D. Unit. Characteristics of atmospheric turbulence near the ground, part iii: variations in space and time for strong winds (neutral atmosphere). Technical report, 2001.

[105] H. Utsunomiya, F. Nagao, M. Noda, and E. Tanaka. Vortex-induced oscillation of a bridge in slowly fluctuating wind. *J. Wind Eng. Ind. Aerodyn.*, 89:1689–1699, 2001.

[106] T. Vejrum, D. J. Queen, G. L. Larose, and A. Larsen. Further aerodynamic studies of Lion' gate bridge - 3 lane renovation. *J. Wind Eng. Ind. Aerodyn.*, 88:325–341, 2000.

[107] T. von Kármán and W. R. Sears. Airfoil theory for non-uniform motion. *J. Aero. Sci.*, 5(10):379–390, 1938.

[108] J. H. Walther and A. Larsen. Discrete vortex method for application to bluff body aerodynamics. *J. Wind Eng. Ind. Aerodyn.*, 67–68:183–193, 1997.

[109] F. M. White. *Viscous Fluid Flow.* McGraw Hill, Inc., 2. edition, 1991.

[110] J. H. Williamson. Low-storage Runge-Kutta schemes. *J. Comput. Phys.*, 35:48–56, 1980.

[111] G. Winckelmans, R. Cocle, L. Dufresne, and R. Capart. Vortex methods and their application to trailing wake vortex simulations. *C. R. Physique*, 6:467–486, 2005.

[112] G. S. Winckelmans and A. Leonard. Contribution to vortex particle methods for the computation of three-dimensional incompressible unsteady flows. *J. Comput. Phys.*, 109:247–273, 1993.

[113] J. C. Wu. Numerical boundary conditions for viscous flow problems. *AIAA J.*, 14(8):1042–1049, 1976.

[114] J. C. Wu and U. Gulcat. Separate treatment of attached and detached flow regions in general viscous flows. *AIAA J.*, 19(1):20–27, 1981.

[115] J. C. Wu and N. L. Sankar. Aerodynamic force and moment in steady and time-dependent viscous flows. In *18th AIAA Aerospace Sciences Meeting*, Pasadena, California, January 1980.

[116] Z. Zhou, A. Chen, and H. Xiang. Numerical assessment of aerodynamic derivatives and critical windspeed of flutter of bridge decks by discrete vortex method. *J. Vib. Engng.*, 15(3):327–331, 2002.

Appendix A

ESDU atmospheric turbulence spectra

The Engineering Sciences Data Unit (ESDU) [103] provides spectral densities of the horizontal and vertical atmospheric turbulent fluctuations, S_{uu} and S_{ww} respectively. Parameter values used in the present work are stated as the parameters are introduced.

$$\frac{nS_{uu}}{\sigma_u^2} = \beta_1 \frac{2.987 n_u/\alpha}{\left[1 + (2\pi n_u/\alpha)^2\right]^{\frac{5}{6}}} + \beta_2 \frac{1.294 n_u/\alpha}{\left[1 + (\pi n_u/\alpha)^2\right]^{\frac{5}{6}}} F_1, \tag{A.0.1}$$

$$\frac{nS_{ww}}{\sigma_w^2} = \beta_1 \frac{2.987\left[1 + (8/3)(4\pi n_w/\alpha)^2\right](n_w/\alpha)}{\left[1 + (4\pi n_w/\alpha)^2\right]^{\frac{11}{6}}} + \beta_2 \frac{1.294 n_w/\alpha}{\left[1 + (2\pi n_w/\alpha)^2\right]^{\frac{5}{6}}} F_2, \tag{A.0.2}$$

with the reduced frequencies

$$n_u = \frac{n({}^x L_u)}{U}, \tag{A.0.3}$$

$$n_w = \frac{n({}^x L_w)}{U}. \tag{A.0.4}$$

Tabulated values are used for the coefficients α, β_1 and β_2 cf. [42]. In the present work these are $\alpha = 0.662$, $\beta_1 = 0.80$ and $\beta_2 = 0.20$. The longitudinal

standard deviations

$$\sigma_u = \frac{7.5\eta u_* \left[0.538 + 0.09 \ln\left(\frac{z}{z_0}\right)\right]^2}{1 + 0.156 \ln\left(\frac{u_*}{f z_0}\right)}, \tag{A.0.5}$$

$$\sigma_w = \sigma_u \left[1 - 0.45 \cos^4\left(\frac{\pi}{2}\frac{z}{h}\right)\right], \tag{A.0.6}$$

where

$$\eta = 1 - \frac{6fz}{u_*}, \tag{A.0.7}$$

$$p = \eta^{16}, \tag{A.0.8}$$

depend of the friction velocity u_* from the logarithmic law profile

$$U(z) = \frac{u_*}{k} \ln\left(\frac{z}{z_0}\right). \tag{A.0.9}$$

k is the von Kármán constant assumed to be 0.4, z_0 is the surface roughness length. u_* is determined from the velocity $U = 35\,\mathrm{m/s}$ at $z = 70\,\mathrm{m}$ above ground level when $z_0 = 0.003\,\mathrm{m}$ for open water is used [104]. $f = 10^{-4}\,\mathrm{Hz}$ is the mid-latitude Coriolis frequency and $h = 660\,\mathrm{m}$ the atmospheric boundary layer thickness. The longitudinal length scales of the horizontal and vertical fluctuations are given by

$${}^{x}L_u = \frac{A_k^{\frac{3}{2}} \left(\frac{\sigma_u}{u_*}\right)^3 z}{2.5 K_z^{\frac{3}{2}} \left(1 - \frac{z}{h}\right)^2 \left(1 + 5.75\frac{z}{h}\right)}, \tag{A.0.10}$$

$${}^{x}L_w = {}^{x}L_u \left[0.5 \left(\frac{\sigma_w}{\sigma_u}\right)^3\right], \tag{A.0.11}$$

and with the present configuration ${}^{x}L_u = 229.6\,\mathrm{m}$ and ${}^{x}L_w = 21.8\,\mathrm{m}$. Also

$$A_k = 0.115 \left[1 + 0.315 \left(1 - \frac{z}{h}\right)^6\right], \tag{A.0.12}$$

$$K_z = 0.19 - (0.19 - K_0) \exp\left[-B_k \left(\frac{z}{h}\right)^{N_k}\right], \tag{A.0.13}$$

$$K_0 = \frac{0.39}{\mathrm{Ro}^{0.11}}, \tag{A.0.14}$$

$$B_k = 24\mathrm{Ro}^{0.155}, \tag{A.0.15}$$

$$N_k = 1.24\mathrm{Ro}^{0.008}, \tag{A.0.16}$$

$$\mathrm{Ro} = \frac{u_*}{f z_0}, \tag{A.0.17}$$

$$F_1 = 1 + 0.455 \exp\left[-0.76 \left(\frac{n_u}{\alpha}\right)^{-0.8}\right], \tag{A.0.18}$$

$$F_2 = 1 + 2.88 \exp\left[-0.218 \left(\frac{n_w}{\alpha}\right)^{-0.9}\right]. \tag{A.0.19}$$

Appendix B

Analytic and discrete Fourier transform

Discretising the Fourier transform is straightforward but reveals details of the discrete Fourier transform (DFT) that must be considered during numerical implementation. The fast Fourier transform (FFT) is a family of efficient DFTs and the present work utilises only FFTs. Therefore the acronym FFT is used throughout even if the content is not specific for the FFT. The FFT is a powerful, but sensitive tool and though erroneous implementation may lead to very large errors they may also not manifest themselves until the error convergence is studied at high numerical resolution.

Let $x(t)$ be a real valued function, then the analytic Fourier transform is given by

$$X(f) = \mathcal{F}(x(t)) = \int_{\infty}^{\infty} x(t) e^{-i2\pi tf} dt \tag{B.0.1}$$

and the inverse Fourier transform by

$$x(t) = \mathcal{F}^{-1}(X(f)) = \int_{\infty}^{\infty} X(f) e^{i2\pi tf} df. \tag{B.0.2}$$

Both are unitary, i.e. no normalisation is necessary. The one-sided spectral density S_{xx}, or merely spectrum, is defined as

$$S_{xx}(f) = 2X(f)X(f)^*, \quad f \in [0; \infty], \tag{B.0.3}$$

where * denotes the complex conjugate. The spectral density is expressed in rad/s as

$$S_{xx}(\omega) = \frac{1}{2\pi} S_{xx}(f), \tag{B.0.4}$$

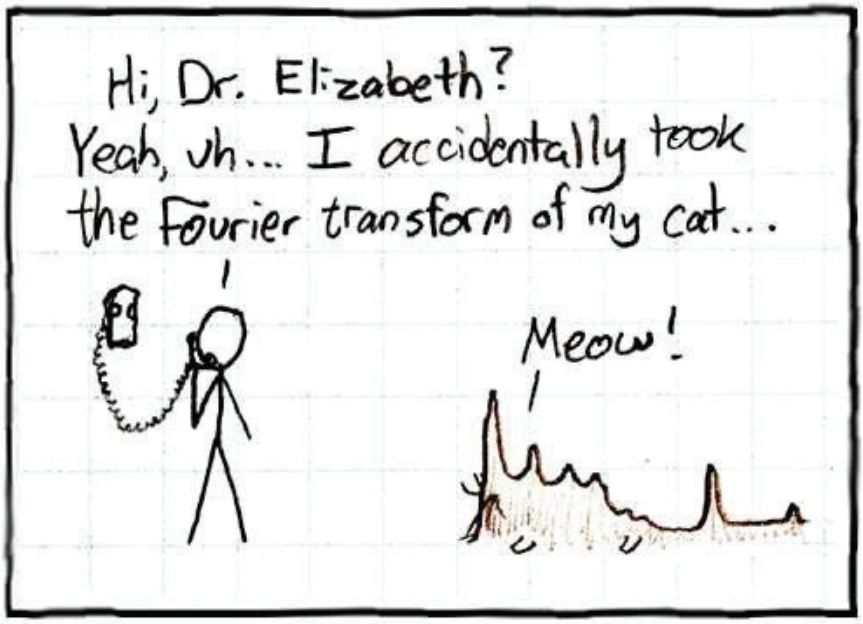


Figure B.1: The Fourier Transform can be tricky. From xkcd.com

such that the integral of the one-sided spectrum is equal to the variance of $x(t)$

$$\sigma_x^2 = \int_0^\infty S_{xx}(\omega) d\omega. \tag{B.0.5}$$

Eq. (B.0.1) can be written in discrete form as

$$X(f_k) = \sum_{j=0}^{N-1} x(t_j) e^{-i2\pi t_j f_k} \Delta t \tag{B.0.6}$$

where the signal $x(t)$ is periodic with the period T as . Then

$$\Delta t = \frac{T}{N} \tag{B.0.7}$$

$$f_{max} = \frac{1}{\Delta t} \tag{B.0.8}$$

$$\Delta f = \frac{f_{max}}{N} = \frac{1}{\Delta t N} = \frac{N}{TN} = \frac{1}{T} \qquad \Rightarrow \qquad \Delta f = \frac{1}{T} \tag{B.0.9}$$

$$t_j = \Delta t j = \frac{Tj}{N} \qquad \Rightarrow \qquad t_j = \frac{Tj}{N} \tag{B.0.10}$$

$$f_k = \Delta f k = \frac{k}{T} \qquad \Rightarrow \qquad f_k = \frac{k}{T} \tag{B.0.11}$$

The discrete Fourier transform becomes

$$X(f_k) = \frac{T}{N}\sum_{j=0}^{N-1} x(t_j)e^{-\frac{2\pi ijk}{N}} \tag{B.0.12}$$

and the inverse, Eq. (B.0.2),

$$x(t) = \mathcal{F}^{-1}(X(f)) = \int_{\infty}^{\infty} X(f)e^{i2\pi tf}df \tag{B.0.13}$$

$$x(t_j) = \sum_{k=0}^{N-1} X(f_k)e^{i2\pi t_j f_k}\Delta f \tag{B.0.14}$$

$$x(t_j) = \frac{1}{T}\sum_{k=0}^{N-1} X(f_k)e^{\frac{2\pi ijk}{N}}. \tag{B.0.15}$$

The discrete time and frequency scales are defined as

$$t_j = \frac{T}{N}j, \qquad \text{for } j \in [0; N-1] \tag{B.0.16}$$

$$f_k = \frac{k}{T}, \qquad \text{for } k \in [0; N-1] \tag{B.0.17}$$

B.1 Non-periodic signals

The FFT will treat any signal as periodic even if the signal is not. If the signal is vertex based on a grid of width L (from first to last vertex) then

$$\Delta t = \frac{T}{N-1} \tag{B.1.1}$$

$$f_{max} = \frac{1}{\Delta t} \tag{B.1.2}$$

$$\Delta f = \frac{f_{max}}{N} = \frac{1}{\Delta t N} = \frac{N-1}{TN} \qquad \Rightarrow \qquad \Delta f = \frac{N-1}{TN} \tag{B.1.3}$$

$$t_j = \Delta t j = \frac{T}{N-1}j \qquad \Rightarrow \qquad t_j = \frac{T}{N-1}j \tag{B.1.4}$$

$$f_k = \Delta f k = \frac{N-1}{TN}k \qquad \Rightarrow \qquad f_k = \frac{N-1}{TN}k \tag{B.1.5}$$

The discrete Fourier transform and its inverse become

$$X(f_k) = \frac{T}{N-1}\sum_{j=0}^{N-1} x(t_j)e^{-\frac{2\pi ijk}{N}} \tag{B.1.6}$$

and the inverse, Eq. (B.0.2),

$$x(t_j) = \frac{N-1}{NT} \sum_{k=0}^{N-1} X(f_k) e^{\frac{2\pi i j k}{N}}. \tag{B.1.7}$$

This different normalisation and coordinate definition is particularly important for vertex based input; when doing derivatives spectrally or using the convolution theorem. For all periodic signals and *cell-centred* data the definitions of the previous section apply.

B.2 MATLAB

MATLAB has a slightly different definition cf. the documentation and to be consistent with the present definitions the FFT commands should be called as

```
X=T/N*fft(x)
x=N/T*ifft(X).
```

The vectors x and X contain no information about the time or frequency scales, hence these must be generated from the definitions B.0.16 and B.0.17 (for periodic signals)

```
t=T/N*(0:N-1)
f=1/T*(0:N-1)
```

or the equivalent discretisation B.1.4 and B.1.5.

B.3 Pure FFTW

MATLAB uses the FFTW library for Fast Fourier Transforms but without normalization on the inverse transform. Thus the results must be scaled as

```
X=T/N*fftw(x)
x=1/T*ifftw(X).
```

The signals and frequencies are given as

```
t=T/N*j,      j in [0 ; N-1]
f=1/T*k,      k in [0 ; N-1]
```

where the last half of the frequency range can be viewed as negative frequencies.

B.4 Derivatives

From Eq. (B.0.2) it is seen that derivatives of x with respect to t are given as

$$\begin{aligned}\frac{\partial}{\partial t}x(t) &= \frac{\partial}{\partial t}\mathcal{F}^{-1}(X(f)) = \int_{\infty}^{\infty}\frac{\partial}{\partial t}X(f)e^{i2\pi tf}df \\ &= \int_{\infty}^{\infty} i2\pi fX(f)e^{i2\pi tf}df \qquad \text{(B.4.1)}\end{aligned}$$

$$\mathcal{F}\left(\frac{\partial}{\partial t}x(t)\right) = \mathcal{F}\left(\mathcal{F}^{-1}(\frac{\partial}{\partial t}X(f))\right) = i2\pi fX(f) \qquad \text{(B.4.2)}$$

i.e. by multiplying the Fourier modes by the corresponding frequency $i2\pi f$.

B.5 Specific for turbulence generation

In case of a prescribed maximum frequency the number of time steps still equals the number of discrete frequencies. For the the time step and the period it implies

$$\Delta t = \frac{1}{f_{max}} \qquad \text{(B.5.1)}$$

$$T = \frac{N}{f_{max}}. \qquad \text{(B.5.2)}$$

If a minimum upper frequency cutoff f_{max} is desired and the time step being is constrained to be a multiple of some value Δt_{DVM}

$$\Delta t = n\Delta t_{DVM}, \qquad n \in \mathbb{Z}^{+} \qquad \text{(B.5.3)}$$

$$f_{max} = \frac{1}{n\Delta t_{DVM}} \qquad \text{(B.5.4)}$$

this also constrains the maximum value that f_{max} can assume cf. Eq. (B.5.4) with $n = 1$.

Appendix C

Finite difference schemes

Derivatives can be approximated numerically in a point from the surrounding grid values by finite difference (FD) schemes. FD schemes are constructed from the Taylor series expansion

$$f(x) = \sum_{n=0}^{\infty} \frac{f^{(n)}(x_0)}{n!}(x - x_0)^n \tag{C.0.1}$$

of a function f at the point x_0, where $f^{(n)}$ denotes the nth derivative of f. If the expansion is truncated after $m + 1$ terms an error of $\mathcal{O}\left((x - x_0)^{m+1}\right)$ is introduced. On an equidistant grid the expansion

$$f(x_i) = \sum_{n=0}^{m} \frac{f^{(n)}(x_e)}{n!}(x - x_0)^n \tag{C.0.2}$$

is evaluated at the grid points $x_i = x_e + \Delta x i$ determined by the index i, where Δx denotes the grid spacing. This can be expressed as a linear system of equations

$$\boldsymbol{f} = \boldsymbol{A}\tilde{\boldsymbol{f}}, \tag{C.0.3}$$

where $\boldsymbol{f}$ is a vector of the $f(x_i)$ values, $\tilde{\boldsymbol{f}}$ is a vector of the derivatives $f(x_e)^{(n)}$ and $\boldsymbol{A}$ is the matrix containing the coefficients of the Taylor series and thus the dependence between the two vectors. Inverting this system yields the derivatives $\tilde{\boldsymbol{f}}$ as weights of the grid values of f. These weights are referred to as finite difference stencils. If x_e is the center grid point the the corresponding stencil is said to be a *centred* finite difference stencil. If is x_e the first or the last points the stencil is based on the stencil is referred to as *one-sided*.

C.1 Extrapolation

By doing the Taylor expansion of f around a point x_e that is the outside the range of points x_i the derivatives are estimated from points disjoint from x_e. The 0th derivative is the function value itself and thus the weights yield the extrapolation of f from the points x_i to the point x_e.

Appendix D

Poisson equation to the stream function

By combining

$$\boldsymbol{u} = \boldsymbol{\nabla} \times \boldsymbol{\Psi} \tag{D.0.1}$$
$$\boldsymbol{\omega} = \boldsymbol{\nabla} \times \boldsymbol{u} \tag{D.0.2}$$

into

$$\boldsymbol{\nabla} \times (\boldsymbol{\nabla} \times \boldsymbol{\Psi}) = -\nabla^2 \boldsymbol{\Psi} + \boldsymbol{\nabla} (\boldsymbol{\nabla} \cdot \boldsymbol{\Psi}) = \boldsymbol{\omega} \tag{D.0.3}$$

one sees that $\boldsymbol{\Psi}$ may be modified by the gradient of some scalar field π: If

$$\boldsymbol{\Psi}' = \boldsymbol{\Psi} + \boldsymbol{\nabla}\pi \tag{D.0.4}$$

then

$$\boldsymbol{\nabla} \cdot \boldsymbol{\Psi}' = \mathbf{0} \tag{D.0.5}$$

by taking the divergence of D.0.4:

$$\boldsymbol{\nabla}^2 \pi = -\boldsymbol{\nabla} \cdot \boldsymbol{\Psi}. \tag{D.0.6}$$

Thus

$$\nabla^2 \boldsymbol{\Psi}' = -\boldsymbol{\omega} \tag{D.0.7}$$

while fulfilling

$$\boldsymbol{u} = \boldsymbol{\nabla} \times \boldsymbol{\Psi}' \tag{D.0.8}$$
$$\boldsymbol{\omega} = \boldsymbol{\nabla} \times \boldsymbol{u}. \tag{D.0.9}$$

Appendix E

Compact vortex rings

Given the stream function in cylindrical coordinates

$$\boldsymbol{\Psi}_{\text{target}} = \mathrm{e}^{-20\,{r_0}^{-1}\left(1-\frac{\rho^2-2\,\rho\,r_0+{r_0}^2+z^2}{{r_0}^2}\right)^{-1}}\,\bar{\mathbf{e}}_\phi, \tag{E.0.1}$$

where $\bar{\mathbf{e}}_\phi$ is the azimuthal unit vector, the corresponding vorticity field is given by applying the Poisson equation

$$\begin{aligned}
\boldsymbol{\omega}_{\text{target}} =& -\boldsymbol{\nabla}\boldsymbol{\Psi}_{\text{target}} \\
=& -\mathrm{e}^{20\,\frac{r_0}{\rho^2-2\,\rho\,r_0+z^2}}\Big(-1600\,z^2{r_0}^2\rho^2-40\,\rho^6 r_0+120\,\rho^5{r_0}^2 \\
&-160\,\rho^4{r_0}^3-8\,\rho^7 r_0+4\,\rho^6 z^2+24\,\rho^6{r_0}^2+6\,\rho^4 z^4-32\,\rho^5{r_0}^3 \\
&+4\,\rho^2 z^6+16\,\rho^4{r_0}^4+160\,\rho^3{r_0}^4-1600\,\rho^4{r_0}^2+3200\,\rho^3{r_0}^3 \\
&-1600\,\rho^2{r_0}^4-80\,\rho^4 r_0\,z^2+80\,\rho^3{r_0}^2 z^2-40\,\rho^2 r_0\,z^4-24\,\rho^5 r_0\,z^2 \\
&+48\,\rho^4{r_0}^2 z^2-24\,\rho^3 r_0\,z^4-32\,\rho^3{r_0}^3 z^2+24\,\rho^2{r_0}^2 z^4-8\,\rho\,r_0\,z^6 \\
&-40\,\rho\,{r_0}^2 z^4+\rho^8+z^8\Big)\left(\rho^2-2\,\rho\,r_0+z^2\right)^{-4}\rho^{-2}\,\bar{\mathbf{e}}_\phi
\end{aligned} \tag{E.0.2}$$

and the corresponding velocity field is found by applying the curl

$$\begin{aligned}
\boldsymbol{u}_{\text{target}} =& \boldsymbol{\nabla}\times\boldsymbol{\Psi}_{\text{target}} \\
=& \,40\,z\,\mathrm{e}^{20\,\frac{r_0}{\rho^2-2\,\rho\,r_0+z^2}}\,r_0\left(\rho^2-2\,\rho\,r_0+z^2\right)^{-2}\bar{\mathbf{e}}_\rho \\
&+\mathrm{e}^{20\,\frac{r_0}{\rho^2-2\,\rho\,r_0+z^2}}\Big(\rho^4-4\,\rho^3 r_0+2\,\rho^2 z^2+4\,\rho^2{r_0}^2-4\,\rho\,r_0\,z^2 \\
&+z^4-40\,\rho^2 r_0+40\,\rho\,{r_0}^2\Big)\rho^{-1}\left(\rho^2-2\,\rho\,r_0+z^2\right)^{-2}\bar{\mathbf{e}}_z,
\end{aligned} \tag{E.0.3}$$

where $\bar{\mathbf{e}}_z$ is the axial unit vector.

Appendix F

Higher order velocity kernels

Given a vorticity field $\boldsymbol{\omega}$ the corresponding velocity field $\boldsymbol{u}$ may be determined by the convolution

$$\boldsymbol{u} = \boldsymbol{K} \circ \boldsymbol{\omega} \tag{F.0.1}$$

where

$$\boldsymbol{K} = -\frac{1}{4\pi|\boldsymbol{x}|^3}\boldsymbol{x}\times \tag{F.0.2}$$

is the singular, three-dimensional, free-space velocity kernel. It may be regularised following the method of [5]

$$\boldsymbol{K}_\sigma = -\frac{q_\sigma(\boldsymbol{x})}{|\boldsymbol{x}|}\boldsymbol{x}\times \tag{F.0.3}$$

where

$$q_\sigma(\boldsymbol{x}) = q(|\boldsymbol{x}|/\sigma) \tag{F.0.4}$$

$$q(\rho) = \int_0^\rho \zeta(t)t^2 dt. \tag{F.0.5}$$

and ζ is a smooth vorticity core, e.g.

$$\zeta^{(2)} = \frac{1}{\sqrt{2\pi}}e^{x^2/2}. \tag{F.0.6}$$

Order p convergence may be obtained by extending the $\mathcal{O}(2)$ core, Eq. (F.0.6), to

$$\zeta^{(p)}(\rho) = \left(\sum_{k=1}^{p/2} c_k \rho^{2k-2}\right) q^{(2)}(\rho). \tag{F.0.7}$$

The coefficients c_k are determined under the constraints of the weight of the being 1 and all even moments up the $p-2$th moment being zero. Velocity kernels up to $\mathcal{O}(10)$ have been constructed, see Table F.1.

$$\boldsymbol{K}_\sigma^{(4)} = \frac{1}{4\pi|\boldsymbol{x}|^3}\left(\frac{1}{\sqrt{2\pi}}\left(-2\frac{|\boldsymbol{x}|}{\sigma}+\left(\frac{|\boldsymbol{x}|}{\sigma}\right)^3\right)e^{-\left(\frac{|\boldsymbol{x}|}{\sigma\sqrt{2}}\right)^2}+\operatorname{erf}\left(\frac{|\boldsymbol{x}|}{\sigma\sqrt{2}}\right)\right)\boldsymbol{x}\times$$

$$\boldsymbol{K}_\sigma^{(6)} = \frac{1}{4\pi|\boldsymbol{x}|^3}\left(\frac{1}{4\sqrt{2\pi}}\left(-8\frac{|\boldsymbol{x}|}{\sigma}+9\left(\frac{|\boldsymbol{x}|}{\sigma}\right)^3-1\left(\frac{|\boldsymbol{x}|}{\sigma}\right)^5\right)e^{-\left(\frac{|\boldsymbol{x}|}{\sigma\sqrt{2}}\right)^2}+\operatorname{erf}\left(\frac{|\boldsymbol{x}|}{\sigma\sqrt{2}}\right)\right)\boldsymbol{x}\times$$

$$\boldsymbol{K}_\sigma^{(8)} = \frac{1}{4\pi|\boldsymbol{x}|^3}\left(\frac{1}{24\sqrt{2\pi}}\left(-48\frac{|\boldsymbol{x}|}{\sigma}+89\left(\frac{|\boldsymbol{x}|}{\sigma}\right)^3-20\left(\frac{|\boldsymbol{x}|}{\sigma}\right)^5+1\left(\frac{|\boldsymbol{x}|}{\sigma}\right)^7\right)e^{-\left(\frac{|\boldsymbol{x}|}{\sigma\sqrt{2}}\right)^2}+\operatorname{erf}\left(\frac{|\boldsymbol{x}|}{\sigma\sqrt{2}}\right)\right)\boldsymbol{x}\times$$

$$\boldsymbol{K}_\sigma^{(10)} = \frac{1}{4\pi|\boldsymbol{x}|^3}\left(\frac{1}{192\sqrt{2\pi}}\left(-384\frac{|\boldsymbol{x}|}{\sigma}+1027\left(\frac{|\boldsymbol{x}|}{\sigma}\right)^3-349\left(\frac{|\boldsymbol{x}|}{\sigma}\right)^5+35\left(\frac{|\boldsymbol{x}|}{\sigma}\right)^7-\left(\frac{|\boldsymbol{x}|}{\sigma}\right)^9\right)e^{-\left(\frac{|\boldsymbol{x}|}{\sigma\sqrt{2}}\right)^2}+\operatorname{erf}\left(\frac{|\boldsymbol{x}|}{\sigma\sqrt{2}}\right)\right)\boldsymbol{x}\times$$

Table F.1: Velocity kernels of order $4, 6, 8, 10$.

Appendix G

Flow visualisations

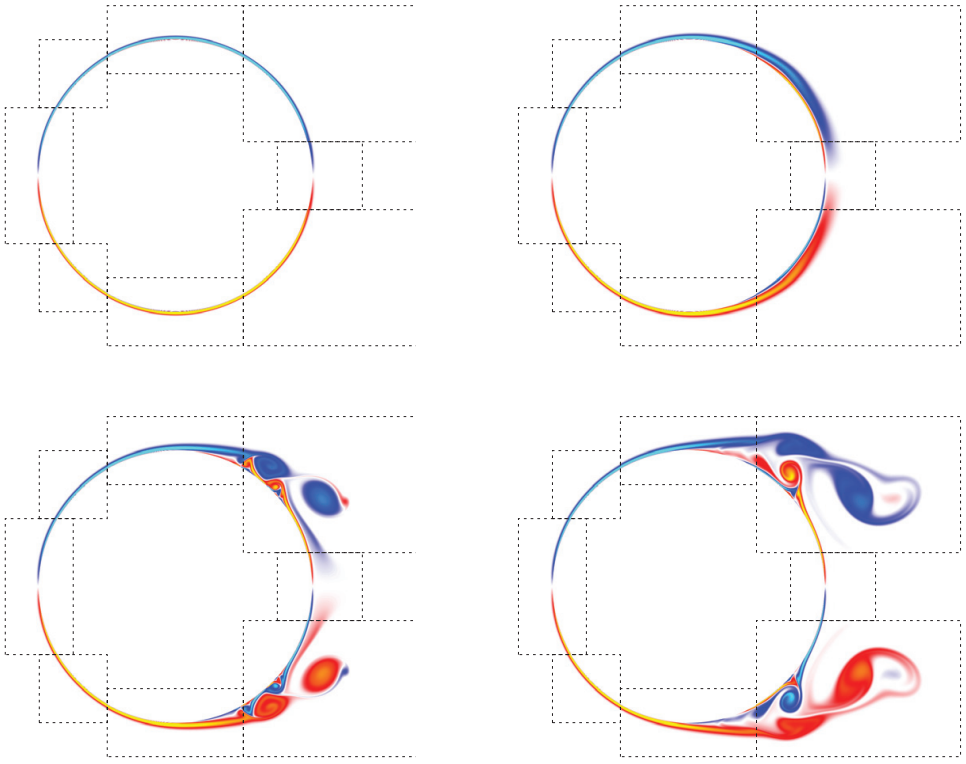


Figure G.1: The vorticity around a cylinder in an onset flow for Re = 9500 at $tU/R = 0.6, 1.4, 2.6, 3.8$. The lines (- - - - -) and (·····) show the patches of resolution $\Delta x_2 = R/256$ and $\Delta x_3 = R/512$. The base patch (level 1) is not shown. Refer to section 3.7.3 for further description.

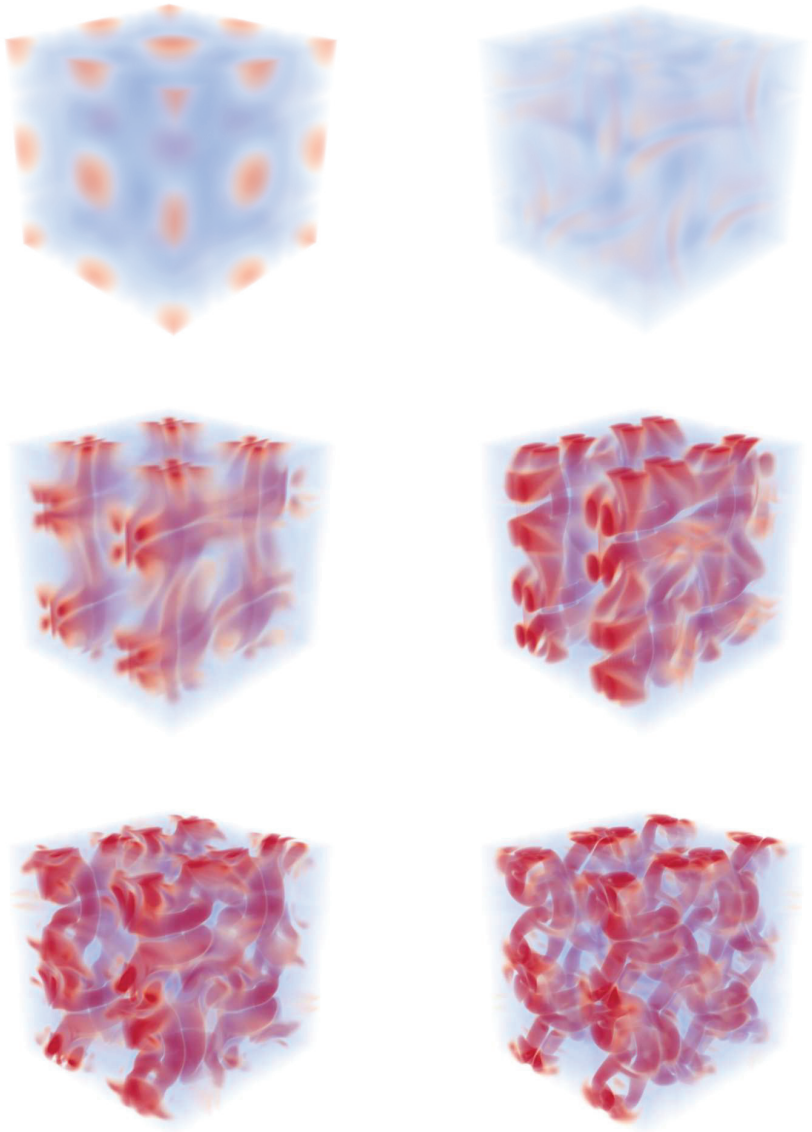


Figure G.2: Vorticity magnitude of Taylor-Green vortices at time $t^* = 0, 2, 4, 6, 8, 10$. Reynolds number 300, $128 \times 128 \times 128$ flow resolution. Refer to section 4.2.4 for further description.

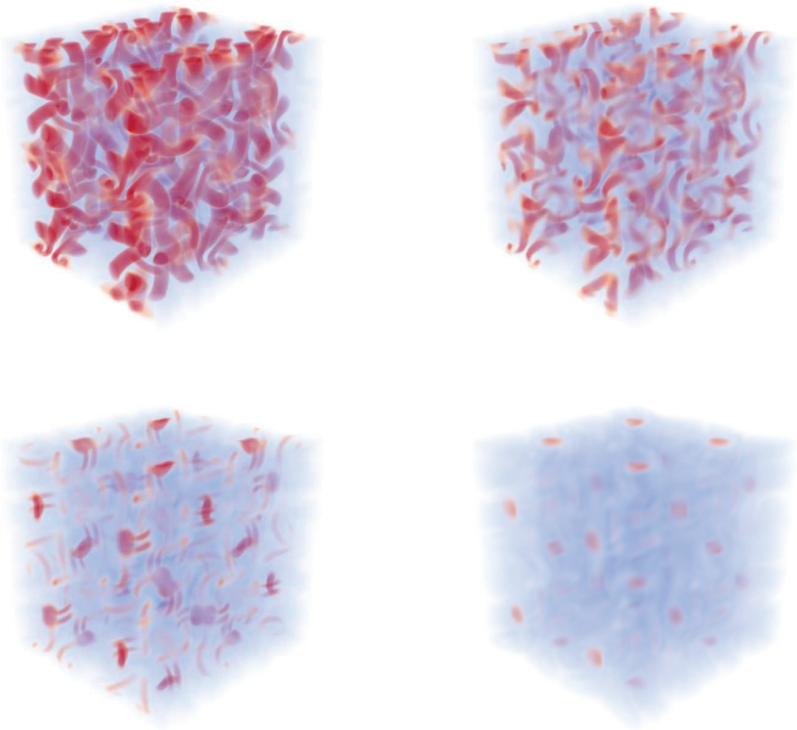


Figure G.2: Continued. $t^* = 12, 14, 16, 18$

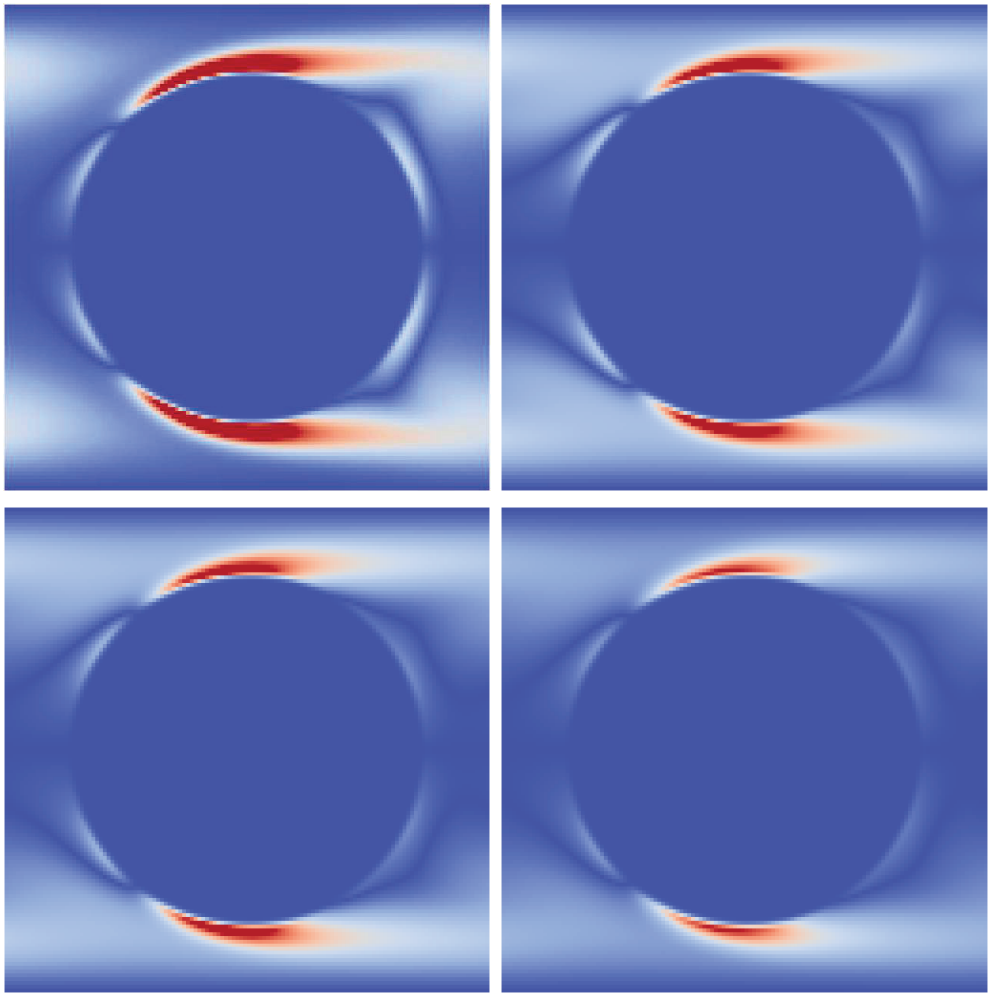


Figure G.3: Flow around a simple cubic array of spheres outlined by (the magnitude of) the boundary layer vorticity. Vorticity magnitude on a cross-section through the center of the domain from 0 (blue) to $|\boldsymbol{\omega}| = 20$ at times $t^* = 1.1, 2.2, 3.3, 12.1$. Refer to section 4.3.1 for further description.

Figure G.4: Top view of the flow around a section of the approach span of the Great Belt bridge at Re=1000. Magnitude of vorticity from blue to red (minimum to maximum) visualised at $t^* = 1.25, 2.5, 3.75$. Refer to section 4.6 for further description.

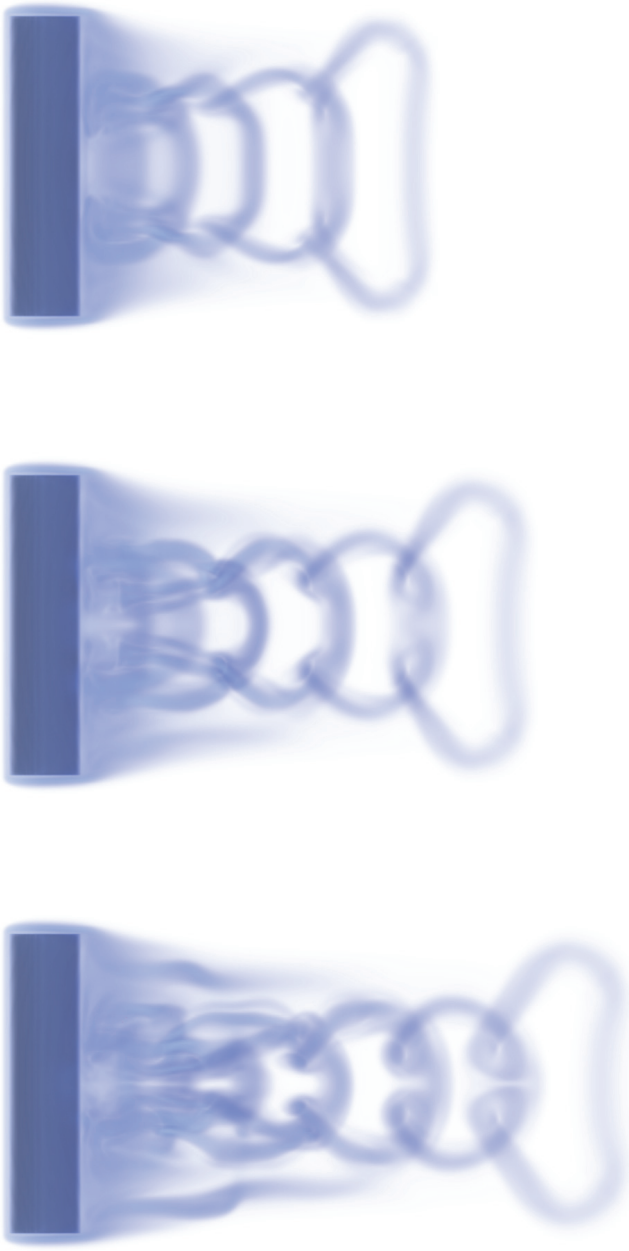


Figure G.4: Continued. $t^* = 5.0, 6.25, 7.5$

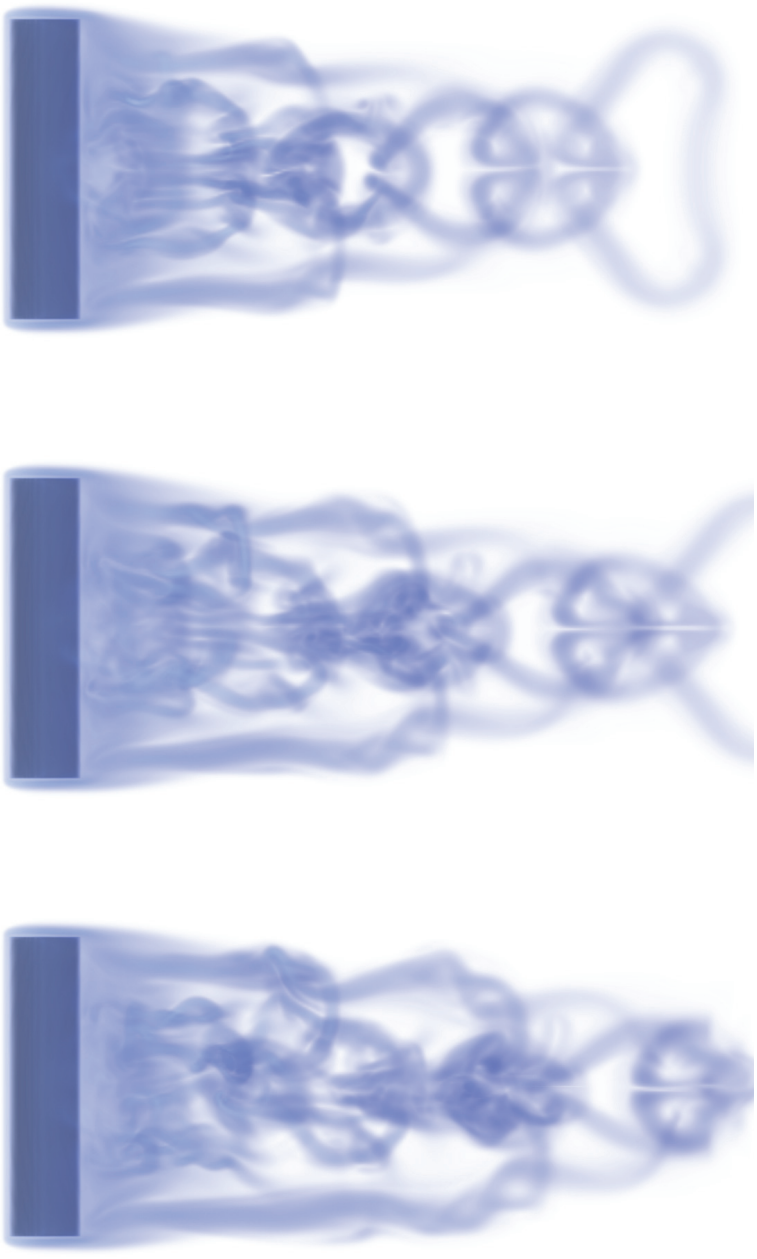


Figure G.4: Continued. $t^* = 8.75, 10.0, 11.75$

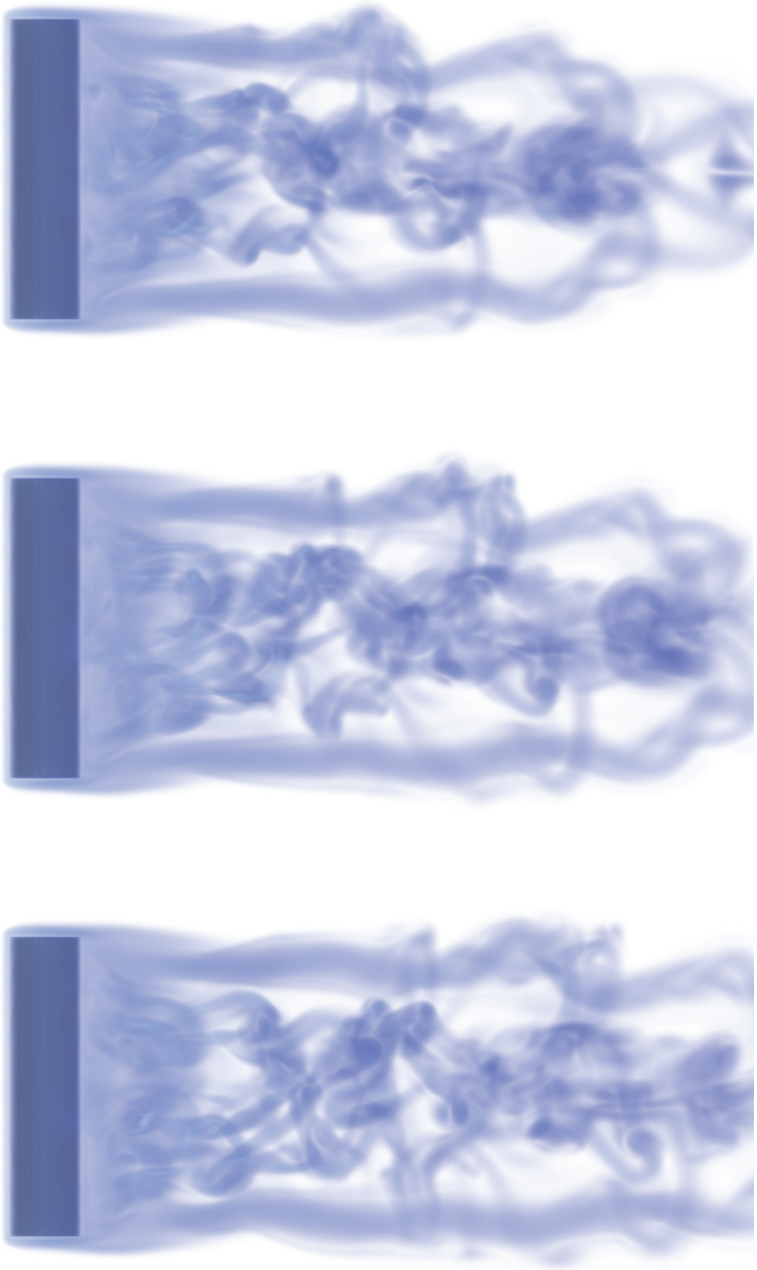


Figure G.4: Continued. $t^* = 12.5, 13.75, 15.0$

DTU Mechanical Engineering
Section of Fluid Mechanics
Technical University of Denmark

Nils Koppels Allé, Bld. 403
DK- 2800 Kgs. Lyngby
Denmark
Phone (+45) 4525 1360
Fax (+45) 4588 4325
www.mek.dtu.dk

ISBN: 978-87-90416-86-7

DCAMM
Danish Center for Applied Mathematics and Mechanics

Nils Koppels Allé, Bld. 404
DK-2800 Kgs. Lyngby
Denmark
Phone (+45) 4525 4250
Fax (+45) 4593 1475
www.dcamm.dk
ISSN: 0903-1685